PROCEEDINGS

NOVEMBER 29- DECEMBER 2, 1993

ARMY
NAVY
USMC
AIR FORCE
INDUSTRY
INTERSERVICE/INDUSTRY TRAINING SYSTEMS AND EDUCATION CONFERENCE

AMERICAN DEFENSE PREPAREDNESS ASSOCIATION
U.S.

PROCEEDINGS

15th INTERSERVICE / INDUSTRY TRAINING SYSTEMS AND EDUCATION CONFERENCE

November 29 - December 2, 1993

Forward

The *Proceedings* of the 15th Interservice/Industry Training Systems and Education Conference (I/ITSEC) contain all papers to be presented. The success of poster displays led us to a specific session allocated to poster papers. This allows authors to provide an in-depth discussion of their research.

This year's papers are presented in six tracks.

Policy and Management
Education, Instruction and Training Methodology
Training, Development and Delivery
Modeling and Simulation
Simulation and Training Systems
R&D Technology Applications

The Conference Committee listed on the following pages devote a great deal of time and effort to make this conference a success and they have my sincere appreciation. Each year we try to present innovative approaches and solutions to current problems. Please share your ideas for future conferences by completing the forms provided in each session.

On behalf of the entire committee we hope you enjoy the conference.

Judith Shellnutt-Riess

Judith Shellnutt-Riess
Program Chair

INTERSERVICE EXECUTIVE COMMITTEE

Brigadier General John F. Michitsch, USA
Commander, US Army Simulation, Training & Instrumentation Command
Orlando, Florida

Captain Christopher L. Addison, USN
Commanding Officer, Naval Air Warfare Center Training Systems Division
Orlando, Florida

Colonel Robert R. Tyler, USMC
Liaison Officer/Program Director, Marine Corps and Systems
Naval Air Warfare Center Training Systems Division
Orlando, Florida

Dr. Robert R. Barthelemy
Product Group Manager, Training System Program Office
Air Force Aeronautical Systems Center
Wright Patterson AFB, Ohio

Dr. Harold F. O'Neil, Jr.
Professor of Counseling and Educational Psychology
School of Education, University of Southern California

INTERSERVICE STEERING COMMITTEE

Chair
Ralph C. Nelson
Special Assistant, US Army Simulation, Training and Instrumentation Command

Carol F. Denton
Head, Instructional Systems Surface Division, Naval Air Warfare Center Training Systems Division

Major Rick Wrenn, USMC
Project Manager, Naval Air Warfare Center Training Systems Division

Stephen Miller
Director, Acquisition Support, US Air Force Training System Program Office

Charles M. West
Deputy Director, Aviation Warfare Training, Chief of Naval Education & Training

John E. Buckley
Chief, Concepts Branch, Office of Deputy Chief of Staff for Training, US Army Training and Doctrine Command

CONFERENCE COMMITTEE

Conference Chair
J.D. (Jack) Drewett

Program Chair
Judith Shellnutt-Riess

Deputy Program Chair
Edward Ward, Hughes Training Inc.

Subcommittee Chairs:
Policy and Management
Ann O'Kennon, EER Systems

Education, Instruction and Training Methodology
J.C. Williams, Loral Advanced Distributed Simulation

Training Development and Delivery
Dr. Lois Wilson, Loral Aerospace

Modeling and Simulation
Dennis Shockley, Evans and Sutherland

Simulation and Training Systems
Ron Johnson, Management Action Corp.

R&D Technology Application
Bob Wells, Lockheed Fort Worth Co.

ADPA Representative
Captain Nelson P. Jackson, US Navy (Ret.),
Director,Technical Services

NTSA Representative
VADM John Disher, US Navy (Ret.), Executive Director

Best Paper Committee
Frank Thweatt, STRICOM

Facilities Committee
Jim Cooksey, Pulau Electronics

Publications Committee
Cheval Force, IBM

Scholarship Committee
G.P.(Pres) McGee, Lockheed Fort Worth Co.

Academic Advisor
Dr. A. Louis Medin, Institute for Simulation and Training

Academic Liaison
Lt. Col. Milt Nielsen, PhD, US Air Force Academy

INSTRUCTIONAL TECHNOLOGY ADVISORY GROUP

Chair
Frank Schufletowski, PhD, US Air Force, Air Education and Training Command

Members
Andy Andrews, Los Alamos National Laboratory
Robert Baudrau, MAST International
Gary Boycan, Office of the Secretary of Defense
John W. Hurlimann, American Academy of Distrance Learning
Francis McCoy, Naval Undersea Warfare Center

POLICY AND MANAGEMENT SUBCOMMITTEE

Chair
Ann O'Kennon, EER Systems

Deputy Chair
Len Kravitz, Reflectone

Members
Stan Aronberg, SIMTEC, Inc.
LTC Ed Aufderheide, NGB/XOO
Bob Bret, CAE-Link Corp.
William L. Brigadier, TRW, Inc.
CDR Jim Clager, Chief of Naval Operations, N889F2
Jim Cooksey, Pulau Electronics
Jay Erwin, Naval Air Systems Command, PMA-205
Gene Holcomb, Harris Corp.
Maj Elaine Howell, HSD/XRTH
George Hoyt, OCI
Dennis Knott, NETPMSA
Dave Manning, STRICOM
Jeff Marlin, Firearms Training Systems, Inc.
Ric Morrow, Martin Marietta
Maj Pat Mullen, USAF
James O'Bryant, STRICOM
Kathy Walsh, SAIC
Jim Williams, Institute for Simulation and Training
Dusty Woodlee, AAI Corp.
Sam Worrell, FAAC, Inc.

EDUCATION, INSTRUCTION AND TRAINING METHODOLOGY SUBCOMMITTEE

Chair
J.C. Williams, Loral Advanced Distributed Simulation, Inc.

Deputy Chair
Elizabeth Armistead, Martin Marietta Info Systems

Members
Bill Beasley, Army Management Engineering College
Gareth Beitzel, AETC/TTDP
LTC Ed Chenoweth, USAF, AWC/TNA
Keith Davis, CTA, Inc.
Dr. Dave Dion, Hughes Training, Inc.
Dr. Steve Goldberg, Army Research Institute (STRICOM)
Doug Greenlaw, Delex Systems, Inc.
Lou Hightower, Computer Sciences Corp.
Steve Husak, Dual Inc.
Dr. Jim McConville, Loral Defense System
Dr. Bill Melton, HQ TRADOC
Ted Pearson, NAWCTSD
Karl Reeves, Quintron
Howard Talley, McDonnell Douglas Training Systems
Tim Toia, IBM Federal Systems Co.
Maj Michael Whyte, PhD, USAFA/CWIS
Dr. Judy Wyne, Chief of Naval Education and Training

TRAINING DEVELOPMENT AND DELIVERY SUBCOMMITTEE

Chair
Dr. Lois Wilson, Loral Aerospace

Deputy Chair
Stephen Stankiewicz, AAI Corp.

Members
Larry Barks, Naval Air Systems Command, PMA -205
John G. Buckley, HQ TRADOC
Larry Clemons, JWK International
Dr. Donna Cherry, JWK International
Dr. Dexter Fletcher, Institute for Defense Analysis
Al Galbavy, ARLHRED (STRICOM)
Katharine Golas, Southwest Research Institute
G. Larry Graham, Grummann Melbourne Division
Lt Col Tom Hawley, USAFA/CWIT
Jim Horton, Loral Defense Systems
John Larson, Boeing Defense & Space Group
Tom Moberg, CAE-Link
Margot Moore, HQ TRADOC
Maj Richard O'Neal, ATC/XPCR
Dr. Harold O'Neil, Educational Psychology & Technology, USC
John Park, Dynamics Research Corp.
Herb Stacy, Chief of Naval Technical Training
Robert Szumowski, Unisys
Jackie Townsend, ASC/YTPJ
Ron Wimberley, Martin Marietta Info Systems

MODELING AND SIMULATION SUBCOMMITTEE

Chair
Dennis Shockley, Evans and Sutherland

Deputy Chair
Joseph Brann, IBM

Members
Jay Anton, Loral Aviation Testbed
Gary Baker, Boeing Defense & Space Group
Art Banman, Concurrent Computer Corp.
Lt Col Dave Bartlett, USMC, Defense Modeling & Simulation
Ben Blood, Coleman Research
Paul Chatelier
Neale Cosby, Institute for Defense Analyses
Earle Denton, E.L. Denton & Associates
Hal Hansen, USA Training Support Center
James Keller, Naval Air Systems Command
Rick Leavitt, Computer Science Corp.
Dr. Edward Martin, ASC/ENST
Don Mathis, Chief of Naval Education and Training
Dr. Bruce McDonald, Coleman Research Corp.
Bruce Montag, Southwest Research Institute
Judith Pafford, Hughes Training Inc.
Ron Schneider, Reflectone
Lundie Sherretz, Scientific Systems Co.
George Snyder, Loral Defense Systems
Phillip Sprinkle, STRICOM
Ed Trier, STRICOM

SIMULATION AND TRAINING SYSTEMS SUBCOMMITTEE

Chair
Ron Johnson, Management Action Corp.

Deputy Chair
Don Jacobs, Pulau Electronics

Members
Mark Adducchio, ASC/YTMS
Dorothy Baldwin, Lockheed Fort Worth Co.
Mike Batsel, SBS Engineering Inc.
Steve Eaton, Chief of Naval Air Training
Bob Epps, CAE-Link Corp.
Frank Lawler, Syntech
Maj James Leahy, STRICOM
Corky Livingston, Analysis & Technology
Jerry McLemore, Naval Air Maintenance Training Group
Hugh Merry, OO-ALC/LIRD
Clint Parker, Metters Industries, Inc.
Cecil Ramsey
Eldon Riley, Rockwell
Scott Royse, Southwest Research Institute
Ron Sheffer, IBM
Bill Simmons, ECC
Bill Walsh, MEI Technology Corp.
Maj Don Walters, ASC/YTCB
Col Michel Whittenberg, DAMO-TRS
Gregg Wood, Star Technologies, Inc.
Maynard Zettler, NAWCTSD

R&D TECHNOLOGY APPLICATION SUBCOMMITTEE

Chair
Bob Wells, Lockheed - Ft. Worth Co.

Deputy Chair
C. Dan Eliason, North American Aircraft

Members
Ken Beckham, Reflectone
Pam Bridges, Chief of Naval Education and Training
Jim E. Brown, ASC/YTEE
James Burke, Harris Corp.
Eric Haseltine, Walt Disney Imagineering
Jerry Hendrix, Boeing Defense & Space Group
William Holtsman, CAE-Link Corp.
Eugene I. Lease, NAWCTSD
Rick Maule, Lockheed Missiles and Space
Ron Muffler, Evans & Sutherland
Gary Nesta, CAE-Link Corp.
Gary Neuman, STRICOM
Bob Parker, General Dynamics/Land System
Dwight Potter, Boeing Defense & Space Group
Randy Saunders, Hughes Training Inc
Dr. Renata Schmidt, GPS Technologies
Doug Smith, HQ, U.S. Army Materiel Command
Dr. Barbara Sorensen, AL/HRMM
Gary Teper, Information Spectrum, Inc.
Stephen Trencansky, Pegasus Real-Time, Inc.
Rick Worel, STRICOM/NAWCTSD

TABLE OF CONTENTS

SOMETIMES HOT, SOMETIMES COLD: WHAT IS THE FUTURE OF MPT & HF ANALYSIS AND PLANNING IN DoD ACQUISITION

Lawrence D. Howell Jr., Ph.D., P.E.
Paul H. Cunningham
University of Dayton Research Institute
Crew System Ergonomics Information Analysis Center
Dayton, Ohio

ABSTRACT

For many years the subjects of Manpower, Personnel, Training, and Human Factors planning oscillated between hot topics to espouse as essential for improved acquisition planning and execution in the Department of Defense, and out-of-favor subjects that caused the eyes of acquisition managers to glaze over. This paper explores how these topics have been treated in the past and how advanced technology and high-level administrative support may lead to improved human-system synergistic performance in the future.

ABOUT THE AUTHORS

Dr. Lawrence Howell has an engineering degree from the Air Force Academy, an M.B.A. from Southern Illinois University, and an M.S. and Ph.D. in Systems Engineering/Operations Research from Ohio State. Dr. Howell has more than 26 years of DoD experience, serving as a pilot, engineer, operations researcher, staff officer, teacher, and manager. He is currently the Associate Director of the Crew Systems Ergonomic Information Analysis Center (CSERIAC).

Paul Cunningham has a B.S. degree in Industrial Technology from Middle Tennessee State University, and an M.S. from the University of Southern California in Systems Management. Mr. Cunningham also has more than 26 years of DoD experience, serving as an aircraft maintenance officer, development engineer, aircraft systems computer simulation engineer, staff officer, and manager. He is currently a Senior Technical Analyst with CSERIAC.

SOMETIMES HOT, SOMETIMES COLD: WHAT IS THE FUTURE OF MPT & HF ANALYSIS AND PLANNING IN DoD ACQUISITION

Lawrence D. Howell Jr., Ph.D., P.E.
Paul H. Cunningham
University of Dayton Research Institute
Crew System Ergonomics Information Analysis Center
Dayton, Ohio

BACKGROUND

The development of computer and other advanced system analysis support equipment, tools, and methods has increasingly made policy decision-making more scientific and cost effective. The purpose of this paper is to assess the trends in Human Systems Integration (HSI) planning and policy making in system acquisition, and to forecast possible changes or advances that may occur over the next decade as a result of improvements in these tools. Specifically, past trends and issues involving Manpower (the number of spaces), Personnel (the number of faces), Training (particular skills and knowledge) and Human Factors (design of system to accommodate the human) in system acquisition will be discussed, along with new tools for Manpower, Personnel, and Training (MPT) analysis, and prospects for integrating MPT and Human Factors (HF) into future system design and analysis.

Past Practices

As far back as 1960 the Services were concerned about the lack of tools and processes for addressing human issues in system acquisition. Many military systems were being developed that did not adequately integrate the person into the design, or consider the people cost to support a system. Weapon systems were being delivered without preplanning for needed personnel or training required for successful utilization. Therefore, many systems were unnecessarily difficult and costly to operate and maintain.

Often, and periodically, MPT elements have been addressed separately in what is referred to as "stove pipe" planning. That is, the organizations and people tasked with planning each separate element did not work closely together to integrate their efforts. This resulted in less than ideal planning for the human elements for many systems. As systems were developed, the design was "thrown over the wall" to the logistician for consideration of MPT issues. This process resulted in many MPT factors being "late to need" and not the optimum solution for the system design.

System designs of the past were primarily hardware system design. They were concerned with how fast a system would go, how high it would fly, what bomb load it would carry, what armament it possessed, or how much it weighed, etc. These were truly not "system" designs, in that they gave very little consideration to the number of people, or the training and maintenance required for a system. When the engineers threw their designs over the wall for other System Program Office (SPO) specialists to consider, there was no one on the other side with the expertise to adequately consider the MPT.

SPOs generally didn't have personnel on their staffs with manpower or personnel experience. They frequently relied on their logistics personnel (maintenance NCOs or officers) to address manpower, personnel, or organizational issues, and often these individuals deferred decisions to the using commands. MPT issues were considered "Not my job."

The using command was also not staffed or organized to address MPT issues in a unified and integrated way. The acquisition community at the commands was usually made up of operators, with a few maintainers. Command manpower analysis functions were usually not assigned to work acquisition questions, or asked to participate in analyzing front-end MPT issues for new acquisitions.

Current Situation

The acceptance of Total Quality Management (TQM) and Concurrent Engineering (also known as Integrated Product Development in the Air Force) by the services, cleared the way for manpower, personnel, training, and human factors to become important considerations in system design, rather than step-children. Since Concurrent Engineering relies heavily on the use of empowered teams (composed of a variety of skills: engineering, logistics, manufacturing, etc.) to concurrently and collectively design a system to optimize system performance, supportability, and life cycle cost; it only makes good sense that people with MPT experience should be part of the team. It is now up to management to populate command acquisition teams with the expertise to consider the entire system. The services cannot afford, in this time of reduced budgets, to continue to consider only the operational aspects of a system. They must consider the total cost of a system in dollars and manpower requirements. In this era of increased competition and cost awareness, it is imperative that "people planning" play a larger part in acquisition process teams and early system design planning.

The consideration of a system's total Life Cycle Cost (LCC) is especially critical as our total military infrastructure is down sized. Also, the importance of adequately considering MPT issues is highlighted since it has been well established that MPT factors account for between 50 and 65 % of most aircraft systems Operations and Support (O&S) cost. Figure 1 (extracted from the Air Force Cost Center) reflects a typical F-16 24 aircraft squadron O&S cost for one year (Dahn, 1992). One year's O&S costs are 53.9M (FY92 dollars). As can be seen, a major portion, 58% or $31.3 million, is dedicated to MPT related cost. These MPT costs are distributed as shown in the stacked bar chart with 77%, or $24.1 million, to train, pay, and provide other direct support to aircraft maintenance personnel every year. As is obvious, there is a great potential for significant dollar savings if more effective ways can be identified, and integrated tools developed, for addressing these MPT issues early in the systems acquisition process.

The importance of making these MPT decisions early in the system acquisition process cannot be overemphasized. As has been demonstrated on numerous acquisitions, approximately 70% of a system's cumulative life cycle cost is set by the time the system reaches Milestone I (the point in the acquisition cycle where a new system is

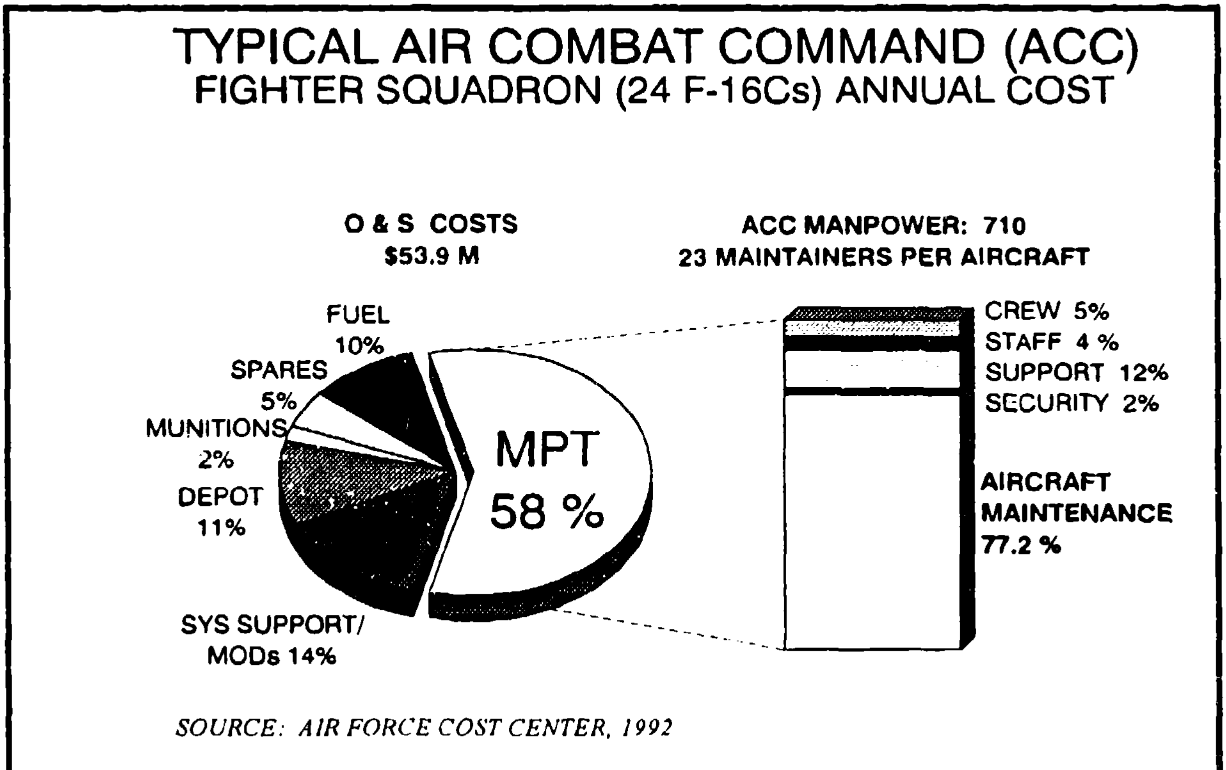


Figure 1: Annual Life Cycle Costs for Typical Air Force Fighter Squadron

moved from concept exploration and definition to demonstration and validation). Figure 2 illustrates the gains possible by addressing acquisition issues early. (Potempa, Gentner, 1990) Along with the previous fact that MPT is responsible for 50 to 65% of a system's cost, the ability to address MPT early is doubly critical. In trying to reduce a system's overall cost (as shown in Figure 1) an obvious place to start is with manpower, especially since maintenance manpower (as can be seen in the above F-16 example 77% of all F-16 manpower is maintenance manpower) accounts for a significantly large portion of the overall system cost.

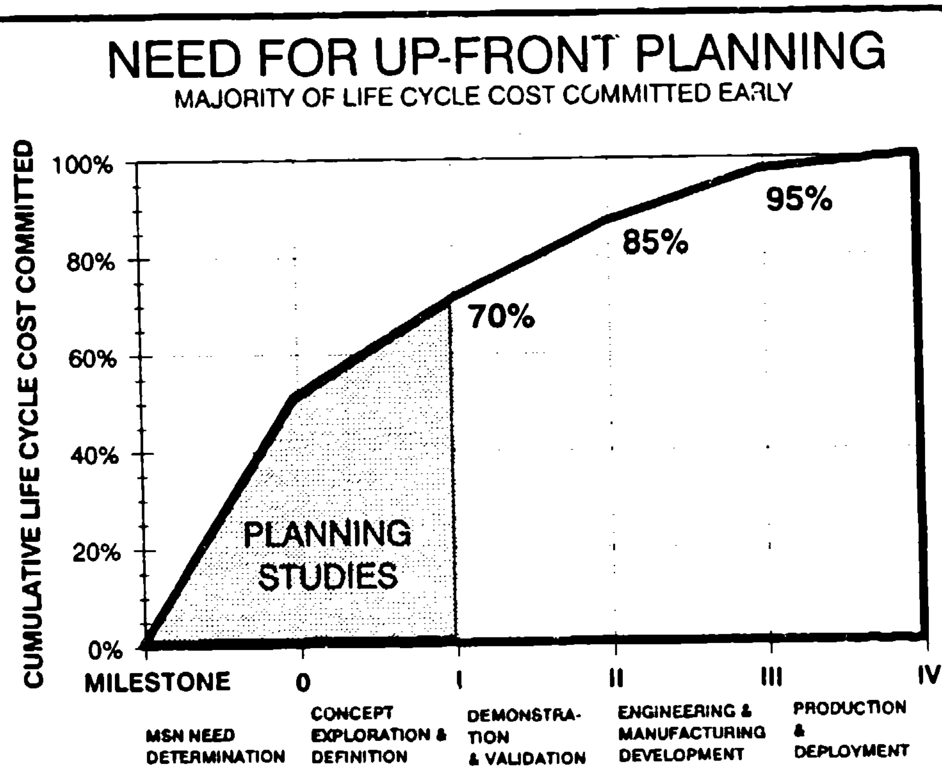


Figure 2: Percentage of Life Cycle Cost Committed by Acquisition Phase

Even though manpower is the obvious area in which to concentrate to lower cost, reducing manpower must be done carefully and with considerable thought and planning, since maintenance manpower requirements are generally driven by a variety of factors. In the above aircraft example, the number of aircraft, sortie rate, organizational structure, reliability, and number of specialties required, are a few of the major factors. The number of specialists a system requires, along with the organizational structure under which they work, are the key areas driving manpower requirements in the above F-16 example. (Cunningham, 1991) The reliability of new and proposed weapon systems (even though it is not the key driver) is such that it is possible to reduce the number of these specialties. Many specialists find themselves idle much of the time because the systems they work, are so reliable they very rarely break (the lonesome Maytag repairman syndrome).

For example, studies in the Air Force suggest that a reduction in the number of specialties, from 14 to 4 or 5, could be made for new systems on the drawing board. Even though this could result in considerable savings in maintenance manpower, it could also create other problems. If one specialist is now responsible for what used to take two specialists, will the one specialist be capable of absorbing and maintaining currency on the information necessary to do the two jobs? Under such conditions, the technician may need to be provided with some type of job aid (either on or off the system, electronic or manual) or additional job training, up-grade training, or refresher training to keep current.

Because of the large amount of training required for maintenance specialist to stay current, the complexity of the training, and the use of computers in training and on the job, Human Computer Interface has become a current area of extreme interest. Technical skills required to operate and maintain systems are becoming less dependent on mechanical aptitude and more dependent on computer skills and abilities. The human factors of designing complete workstations is vitally important to ensure productivity of the human machine interface. Much is left to be done in what remains a key area for future research development.

DoD MPT PROGRAMS

As outlined in the above background discussion Manpower, Personnel, Training, and Human Factors issues are at the heart of reducing manpower requirements and cost for current and future weapon systems. With this understanding DoD established the Human System Integration (HSI) program. At the Department of Defense level the program is referred to as HSI, but the individual services programs supporting the DoD initiative have individual names. The Air Force program is the Integrated Manpower, Personnel, And Comprehensive Training and Safety (IMPACTS) program: the Army calls theirs the Manpower and Personnel Integration (MANPRINT) program; and the Navy prefers Navy Human Systems Integration. Even though the services' programs had similar goals, they have followed very different development and management courses. The Army MANPRINT program was supported and directed from high in the Army chain of command and received "top down" direction and funding, whereas the Air Force effort was a grass-roots program, designed to establish working relationships with SPOs and determine "from the bottom up" what was required in the way of MPT.

Even though the service MPT organizations have achieved a margin of success, they have experienced considerable difficulty in a number of areas. One of the most difficult problems the services' experienced, and the one that has been impossible for the Air Force to overcome, has been the lack of consistent support, both financial and policy. The OPR for the model Air Force organization, at its establishment, was SAF/AQ (Acquisition) but, within the next few years the responsibility shifted from AQ to DP (Personnel), to MO (Manpower) and back to AQ. At the same time, and at a much more rapid pace than the office symbol changes, the general officers responsible for providing high-level guidance and support came and went with lightning speed. The rapid leadership changes and failure to achieve a firm leadership stand caused delays in program development. Each time a general officer OPR would make a commitment to HSI, he would be moved to another assignment and his replacement would have a new agenda, and would not necessarily follow up on commitments made by his predecessor. This constant turnover in leadership resulted in years of fluctuating support. Following a period of increased emphasis in HSI planning in the late 80s and very early 90s, the past two years have again seen a falling off of interest by leadership (Howell, 1989) as a result of recent force down sizing. This lack of concern for HSI has produced the usual result. The July 26-August 1, 1993 issue of the Defense News identified that the last AIM-129A Advanced Cruise Missile is expected to be delivered to the Air Force in August 1993, but no one will be trained to work on it for another two years.

The overriding concern with force down sizing (especially without having the time to plan and execute an optimal approach to force changes, given changing missions, new systems, and new organizational structures) has again caused leadership at nearly all levels to reduce concern for HSI issues. This is ironic since HSI, and the tools being developed for HSI analysis, have as one of their major features the ability to analyze and optimize the manpower requirements for a system, or force structure, as a result of change. It appears the breakneck speed of the draw down and reorganization has allowed only time for "meat cleaver" analysis. This is rather appalling, considering that if management had provided the timely leadership needed, analysis organizations could have provided well thought-out assessments of manpower requirements based on mission and system requirements. These studies could have detailed ways of savings manpower instead of deleting squadrons of aircraft, ships, and tanks (and people) to satisfy manpower draw downs. Although the current frenzy of force draw downs and funding reductions has resulted in reduced support for the active HSI programs, these same factors are likely to bring increased high-level management interest once the short-term reaction to force structure changes has passed. This will likely come about as high-level management is made aware of the manpower and cost savings that can result from the integrated use of newly available analysis tools, along with systems efficiency improvements that follow operator and maintainer workload reductions. In the face of decreased funding, it will become essential to reduce manpower requirements to the absolute minimum to support both new and modified systems.

The services have never been thrilled with the idea of reducing manning. To overcome their resistance, the services will have to be convinced the savings can be made without harming their ability to accomplish their missions. If the services don't accept this idea, then further reductions probably won't take place. This is not to say that the services wouldn't prefer systems that could be

supported with less manpower, it's just that they don't want to give up the manpower they currently have. Manpower means flexibility. Flexibility in war could mean the difference between success and defeat. Therefore, it should not be too surprising that the services have shown little inclination to make manpower reductions. From the users' point of view, if projected reliability in systems, or stability of the world environment doesn't work out, as has happened in the past, they will have to live with reduced manpower, and the resulting problems, for years. (Cunningham, 1991)

The lag between need and procurement is not just a condition confined to obtaining manpower. The time it has taken to develop HSI policy, procedures, and tools has also been a major problem for HSI practitioners. In the late 80s when interest in MPT was at a peak, the services were instructed to establish MPT organizations and to develop policies, procedures, and tools to be able to perform appropriate MPT analysis. All the services struck out smartly to comply with these requirements, but were soon to learn that change is slow in coming. Regulations, policies, handbooks, training, and tools required to do effective MPT analysis have lagged considerably behind management interest. (See Figure 3) When interest was extremely high the MPT community had no tools, guidance, training, or money. As these items have been developed and MPT analysts trained, management interest has moved on to other things. Now that HSI policy is law, and tools are coming on line, management interest is again starting to ramp up, but now without funding. Maybe someday we will get it all together.

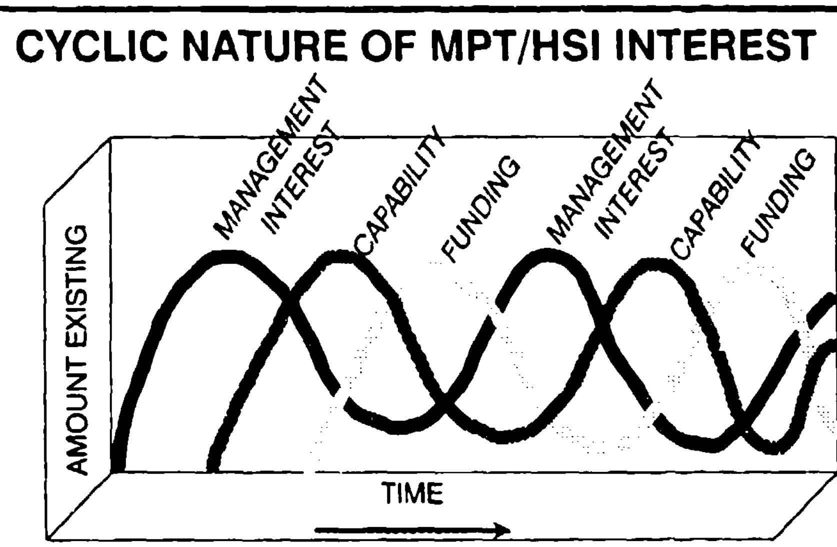


Figure 3: Lag Between Management Interest, Capability, and Funding

REGULATORY GUIDANCE

When briefed on HSI, generally everyone considers the idea of increased emphasis on Manpower, Personnel, Training, and Human Factors "motherhood and apple pie," but they also concede they need lots of help in how to do MPT analysis. In recognition of this need, the Department of Defense (DoD) mandated a series of system acquisition directives, DoDD 5000.1, instructions DoDI 5000.2, and manuals DoD 5000.2M which require HSI analyses during the acquisition process. *Defense Acquisition Management Policies and Procedures*, "...requires the effective integration of human considerations in system design in order to improve total system performance." Design objectives for human components of a system are to be established at Milestone I, then subsequently addressed and refined at each phase of the acquisition process. Further, the Director of Defense Research and Engineering (1992), lists "human-systems interface, design automation, and environmental effects" as three of the eleven key DoD technologies in system design. (Gentner, Crissey, 1993)

TOOLS

With the DoD mandate to perform Human Systems Integration analysis on acquisition systems, government and contractors have produced (or are producing) an impressive array of tools to meet the analysis needs. With this wealth of HSI technologies, it is often difficult to identify which tool is available and appropriate. Under the sponsorship of the Office of the Assistant Secretary of Defense and the North Atlantic Treaty Organization (NATO) Research Study Group 21, the term "Liveware" was coined and a survey conducted to collect and catalog the human-related Manpower, Personnel, Training, Safety, Health Hazard Prevention, and Human Factors Engineering tools. A total of over 500 tools were identified and analyzed. (Gentner, Crissey, 1993) The greatest number of technologies were in the training area, and the fewest in Health Hazards. Over half of the technologies were developed by the military, the remaining by industry and academia. Two of the most impressive MPT tools cataloged belonged to the Air Force and the Army.

The Air Force is developing a Manpower, Personnel, and Training in Acquisition Decision Support System (MPT DSS). The MPT DSS is an integrated set of analysis tools to help inject design influences in the acquisition and modification of Air Force weapon systems in support of the Air Force's IMPACTS program. These tools consist of a Specialty Structuring tool to structure jobs from the ground up, at the task level, or restructure a specialty starting from an existing definition; a Personnel Aptitude and Characteristics model to ensure that the collection of job tasks does not require unreasonably high aptitude levels or physical profile characteristics that can't be supported by the Air Force population; a Training Resources Requirements tool to project an estimate of resources needed to establish and maintain the training pipeline; a Manpower estimating tool to determine the number of people required to operate, maintain, support, and train a single unit or squadron; a Force Structuring tool to aggregate the manpower estimates into wings, groups, MAJCOMS, and force level projections using appropriate overhead and support ratios; an Inventory Projection/Civilian Availability model to determine whether the civilian populace can support the level of aptitude in the numbers identified from the Force Projection model; a Trade-off model to balance between Manpower, Personnel, and Training as they each affect the other. Finally, all decisions will be run through a Life-Cycle Costing model to determine a bottom line dollar figure for the MPT element of the equation.

The Army in support of it's HSI effort has produced a suite of six software programs called HARDMAN III. The six HARDMAN III tools are illustrated in Figure 4. The first module, the System Performance and RAM (Reliability, Availability, and Maintainability) Criteria Estimation Aid (SPARC), is used to set realistic system and mission performance criteria through the use of task network modeling. The Manpower Constraints Estimation Aid (M-CON), Personnel Constraints Estimation Aid (P-CON), and the Training Constraints Aid (T-CON) are used to identify the number of soldiers, and their skills and abilities, and the training resources likely to be available. The last two models, Manpower-based System Evaluation Aid (MAN-SEVAL) and Personnel-based System Evaluation Aid (PER-SEVAL), are used to evaluate system design with respect to the manpower crew size and personnel characteristics required. (Alender, McAnulty, 1992)

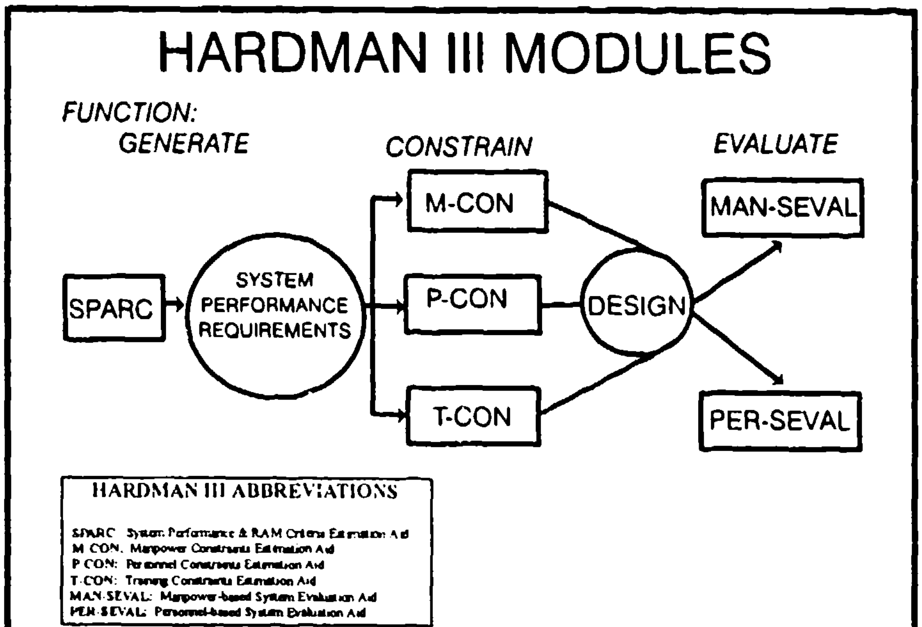


Figure 4: Flow Diagram of HARDMAN III Modules

RECOMMENDATIONS

To recommend change, or advocate change, is generally not a very popular thing to do. Many people don't like advocates, or advocacy programs. They feel that if something's value isn't self evident, then it isn't worth doing. It would be extremely simple to get things done if everyone immediately understood the value of an action, and would independently do whatever was required to accomplish the action. But this isn't the way people behave. Too many times we act like sheep waiting for someone to show us the way to the barn. President John Kennedy was the advocate for putting a man on the moon. Through his vision, we as a nation were able to see.

The following recommendations on how to improve the consideration of HSI issues within the DoD are solely the opinions of the authors and do not reflect the thoughts or recommendations of any other person, service, or organization. The recommendations are based on the authors' fifty two years of combined Department of Defense experience in the areas of manpower, personnel, training, human factors, logistics, operations, acquisition, maintenance, operations research, computer simulation, and requirements policy.

In order for HSI to have the maximum impact on acquisition and cost, there must be a strong advocate, high in the chain of command, who will demand that HSI be given attention. The Secretary of Defense's focal point for Human Systems Integration, Personnel and Readiness (formerly called Force Manpower and Personnel (FM&P)), should be augmented with representatives from all the services, and experts from the HSI domains (Manpower, Personnel, Training, Human Factors, Safety, and Environmental) and empowered as the joint services HSI office for developing policy and procedures. Too often the services have, in their rush to respond to the needs of a rapidly changing environment, unknowingly duplicated research being carried out by one or more of the other services. Establishing Personnel and Readiness as the joint OPR for HSI would be a first step in developing cooperation in the HSI domain. By providing such an organization the possibility for duplication is reduced, and the development on inner service synergy possible.

The development of joint inter service plans and policy without the teeth to back it up, would only be a paper tiger. To give teeth to the tiger, the HSI plan should be made an exit criteria for the Defense Acquisition Board (DAB). No program would be allowed to transition to the next phase until the plan had been approved.

Not only is policy needed, but the proper tools and techniques must also be available. To ensure adequate HSI analysis tools are available, money must be made available for research and analysis, and training. To accomplish this, a Program Element (PE) should be established for HSI, funded, and the money used to develop training, and the tools and techniques necessary to evaluate and tradeoff HSI issues in system acquisition.

A joint service HSI office for research should be established. This would be an ideal way to avoid possible duplication, and develop inner service cooperation. This office would be tasked with developing new HSI tools and techniques, and integrating those the services currently have.

If the establishment of joint HSI policy and research offices are deemed unworkable, the establishment of a joint services working group and steering committee for HSI issues should be considered. The working group would be responsible for evaluating the services individual HSI programs; making recommendations as to how to take advantage of processes and products already developed; and recommending any new research that might be needed. The steering committee would consider the overall needs of the HSI program, the individual service's needs, and make recommendations on how to integrate the best of each.

CONCLUSION

The "integration" of domains and requirements, such as is accomplished in the Human Systems "Integration" program, is not a new concept, but one very difficult to achieve, or sometimes accept. We are so use to working in a narrow "stove pipe" world it is hard to see beyond our own set boundaries. If we are to succeed and prosper in this very competitive world we must be willing to work together and share ideas. The ideas presented in this paper are nothing more than the utilization of the principles of Concurrent Engineering and Total Quality Management. Multi-disciplined teams have been shown to be an excellent way to exchange and perfect ideas. By forming joint service teams to address HSI issues and research, we can produce the most cost effective, and integrated approach for addressing the human in acquisition.

In the classic tale *The Wizard of Oz*, the road to the magic city of Oz was a well marked yellow brick road. As Dorothy and her friends traveled the road it took heart, courage, and intelligence to reach their goal. The road to the consideration of human elements in acquisition is not so well marked as the yellow brick road, but just as hazardous. The services must also have the traits of Dorothy and her friends, as well as patience and perseverance, to obtain their goals. The road to consideration of Manpower, Personnel, Training, and Human Factors in acquisition has been eroded by lack of top-level guidance and support, lack of incentives to reduce manpower, lack of money, lack of experienced practitioners, and the lack of integrated analysis tools to perform tradeoff analysis. But times are changing! The drastic draw down in services' manpower and budgets has reemphasized the need to get the absolute most out of the services' most costly resource -- manpower. The lag between management emphasis, policy, and tools has now been narrowed, or in some cases, eliminated. The services now have (or shortly will have) the tools to respond to the call for increased MPT analysis.

As for the likely future of HSI--although current force draw downs and funding reductions have resulted in reduced support for active programs in HSI, these same factors are likely to bring increased high-level management interest once the short-term reaction to force structure change has passed. In the face of decreased funding, it will become essential to reduce manpower requirements to the absolute minimum to support both new and modified systems.

To overcome the long-term resistance to HSI life cycle planning, the major commands and high-level service leadership must be convinced that the savings are real, and that the benefits are the fielding of well planned usable systems. The upturn in HSI interest and support is just around the corner--hang on.

REFERENCES

1. Alender. L. and McAnulty, M.D. (1992, October). Using Simulation to Support Testing: Implications of a HARDMAN III Application. *Military Testing Association Proceedings*.

2. Cunningham, P. H. (1991, October). *Lack of Incentives for Reducing Maintenance Manpower Requirements in System Acquisition* (Unpublished). Wright-Patterson AFB, Ohio: Aeronautical Systems Division, Manpower, Personnel and Training Directorate.

3. Cunningham, P. H. (1990, October). Management Policy Factors That Drive Maintenance Manpower Requirements. *AL INCLUSIVE*. Wright-Patterson AFB, Ohio: Aeronautical Systems Division (ASC/AL).

4. Dahn, D. A. (1992, August). *Manpower, Personnel and Training Decision Support System "The Human Element"* (Unpublished Slide Briefing). Brooks AFB, Texas: Armstrong Laboratory (AL/HRMM).

5. Gentner, F.C. and Crissey, M. J. (1993, June). *Search for Human Factors and HSI Technologies: Results of the DoD Liveware Study* (Unpublished). Wright-Patterson AFB, Ohio: CSERIAC .

6. Howell, L. D. (1989, October). A Comparison of MPT in the Services. *MANPRINT Bulletin, IV* (2).

7. Potempa, K. W. and Gentner, F. C. (1990, October). Manpower Personnel Training and Safety (MPTS) in the Weapon System Acquisition Process (WASP). Proceedings of the Human Factors Society.

Manpower, Personnel, and Training Analysis in Aerospace System Development

Captain David A. Dahn
Dr. H. Barbara Sorensen
Armstrong Laboratory, Human Resources Directorate
Brooks AFB, TX

ABSTRACT

The Manpower, Personnel, and Training (MPT) in Acquisition Decision Support System (DSS) is an Air Force program providing the first integrated tool for addressing MPT requirements during system acquisition and design. New weapon system development and major modifications have historically neglected how our most important and costly resource – people – will maintain and support the fielded system. Inadequate planning for training and deploying the human element has often delayed system operational dates. This DSS will assist acquisition managers and analysts to effectively integrate people issues (numbers, characteristics, proficiency) with equipment (aircraft) early in the acquisition cycle. Acquisition specialists can use the structured analysis approach provided by the MPT DSS to ensure system people costs are affordable, jobs are properly structured, and people are trained prior to the system becoming operational. The MPT DSS is being designed to support the Human System Integration requirements, now directed under DOD Instruction 5000.2.

ABOUT THE AUTHORS

Capt Dave Dahn is the program manager for the MPT DSS in the MPT Integration Branch of the Human Resources Directorate, Armstrong Laboratory, Brooks AFB, TX. Capt Dahn holds an MS in Computer Engineering from the Air Force Institute of Technology and a BS in Electrical Engineering from Arizona State University.

Dr. H. Barbara Sorensen leads the Joint Service Technology Initiatives in the Plans and Programs Division of the Human Resources Directorate, Armstrong Laboratory, Brooks AFB, TX. Dr. Sorensen holds a PhD in Instructional Design and Computer Technology; an MS in Educational Psychology, Measurement, and Statistics; and 17 years of experience in training systems and software engineering.

Manpower, Personnel, and Training Analysis in Aerospace System Development

Captain David A. Dahn,
Dr. H. Barbara Sorensen
Armstrong Laboratory, Human Resources Directorate
Brooks AFB, TX, USAF

INTRODUCTION

New defense systems acquisition or major modifications have historically focused on system costs, schedule, performance, and in recent years - logistics support. Unfortunately, the human element has always been left for last. Human factors experts have made strides to enhance performance and logistics maintenance work, but how we employ our people (recruiting, job descriptions, personnel abilities, training, organizational responsibilities) and their associated costs are often neglected. The opportunities to optimize the human centered elements are enormous. By considering the human capabilities and limitations, beginning with weapon system conceptualization, the human can be eliminated as the factor which currently restricts combat capability; systems can be maintained faster, smarter, and cheaper; people can be trained better, in less time, with higher efficiency; systems can be made safer for the operator, the maintainer, and the non-combat environment. All of this can be achieved by influencing the design of the weapon system to enhance the combat capability through economies of our human resources.[1]

Need for Human Systems Integration in Aerospace Systems

In this era of decreasing defense budgets, each system is coming under increasing scrutiny concerning mission need, system requirements, logistics support, and life-cycle costs (LCCs) by both Congress and the Department of Defense (DoD)[2]. This scrutiny drives our need to economize the way we employ our human resources to achieve the best people-to-system tradeoff we can obtain. Every system requires people to operate, maintain, and support it. An Air Force cost study showed that up to 60 percent of an aircraft's yearly operations and support costs can be directly attributed to manpower, personnel, and training cost elements[3]. As the Air Force's manpower authorizations continue to shrink, and personnel compensation increases, this trend is likely to increase unless new system designs are influenced and adjusted to make the systems easier to operate, maintain, and support with fewer people at existing skill levels. Early identification of manpower, personnel, training, and safety (MPTS) high (cost) drivers, goals, constraints, and issues can provide positive design influences for new weapon systems if properly integrated into the acquisition and engineering process.

Aerospace system developers are constantly striving to exploit new technologies to achieve better, faster, and more powerful defense systems. As a result, the pace of introducing new technologies is threatening to overwhelm the Air Force at a rate never before experienced. Technology is advancing on a broad front and the MPT process of the '90's cannot adequately support the Air Force of the next decade and beyond. Instead of considering the larger issue of "total system performance," acquisition objectives have historically focused on a few variables, maximizing the probability of completing the system's primary mission while minimizing acquisition related costs.

Total system performance is a key new concept in the acquisition directives. DoDI 5000.2 defines the total system to include not only the prime mission equipment, but also the soldier, sailor, airman, or marine who will use or maintain the system, the logistics support structure for the system, and the

other elements of the operational support infrastructure within which the system must operate.

Many acquisition professionals have recognized that the cost of acquiring a new system is dwarfed by the cost of the "total system." These same people also recognized that if we could quantify the operations, support, and training costs of existing systems, we could identify components of systems that historically have presented technological problems since their introduction. Assessing MPT impacts provides one of the best methods of identifying existing costs of operations. By exploiting this knowledge, aircraft designers can use tomorrow's technology to help solve today's problems rather than simply creating more future challenges.

In the past, the solution to technology problems was to employ smarter or more manpower to address the problem until we overcome the technological hurdle. This was application of contemporary organizational theory focused on adapting organizations to their environment. The military then lived with the results until a major pre-planned product improvement occurred or the service retired the weapon system. Congress took note of the ever-increasing use of manpower to support "low-cost" systems and the associated long term life-cycle costs associated with this solution. As a result, it passed a public law (Title 10, United States Code, Section 2434) mandating that the Department of Defense report all costs of new systems during major milestone reviews, specifically addressing manpower costs, before Congress would approve funds for that system. As a result of defense acquisition management reviews and the public law, when the acquisition directives were revised they included requirements for Human Systems Integration (HSI)[4]. The acquisition directives now dictate that "human considerations shall be effectively integrated into the design effort for defense systems to improve total system performance and reduce costs of ownership by focusing attention on the capabilities and

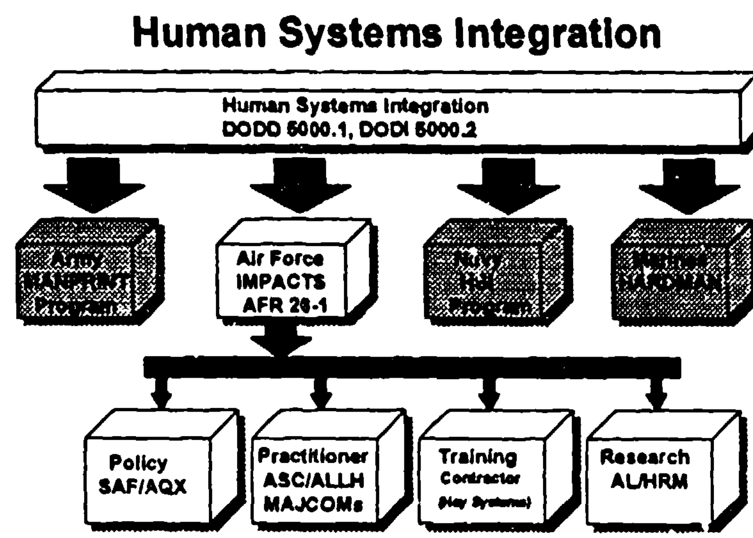


Figure 1

limitations of the soldier, sailor, airman, or marine; and objectives for the human element of the system shall be ... traceable to readiness, force structure, affordability, and wartime operational objectives ...."[5] This level of focus was very problematic for the services because there existed no mechanisms to quantify the costs required or provide the traceability requested. As a result, each military service structured an implementation program (Figure 1) to factor the human into weapon system design — for both new developments or modifications of existing inventory.

The US Air Force's implementation of these procedures is embodied in the Integrated Manpower, Personnel, and Comprehensive Training, and Safety (IMPACTS) Program (AFR 26-1). Just as the Army's Manpower and Personnel Integration (MANPRINT) program, the Navy's Human System Integration (HSI) and Marines' Hardware versus Manpower (HARDMAN) integration program look at integrating soldiers, sailors, and marines into defense systems that are peculiar to those services, the IMPACTS program emphasizes integrating airman into air, space, and ground support defense systems within the Air Force organization.

The IMPACTS program consists of a policy arm, trainers, practitioners, and continuing research (Figure 1) to improve IMPACTS processes and methods. This

paper addresses the research arm of the IMPACTS program and describes an integrated analysis system now being developed. IMPACTS analysts and acquisition managers will use this analysis system to ensure weapon system MPT costs are affordable, jobs are properly structured, and people are trained for their jobs prior to a new system becoming operational.

To achieve the objectives of improved "total system performance" and "reduced costs of ownership" each service needs to consider how the human element interacts with, supports, and is trained to operate and support new technologies and systems. The cost of ownership for today's systems is high; the need for effective MPT analysis in developing new aerospace systems is just as great.

High MPT Operations and Support Costs for Existing Systems

One data source, Air Force Regulation 173-13 — US Air Force Cost and Planning Factors, summarizes operating and support costs for Air Force aircraft. These cost summaries, in conjunction with variable-cost models (e.g., Cost Oriented Resource Estimating (CORE) and the Systematic Approach to Better Long-Range Estimating (SABLE)) are maintained and updated by the Air Force Cost Center, Operating & Support Division (AFCSTC/OS).

Using the CORE model, studies have been done to quantify the MPT portions of yearly operations and support costs. Figures 2 and 3 show the astounding results of one such cost study completed using this AFR 173-13 data[6]. Figure 2 illustrates that up to 62% of the total annual operations and support cost for just one F-16C squadron with 24 primary aircraft authorized is directly attributable to MPT cost elements. The bulk of these MPT costs is due to aircraft maintenance. Figure 3 illustrates the same high MPT costs in the airlift category of aircraft. In this example, 66% of one squadron's annual operations and support costs was directly attributable to MPT cost factors. Summing such costs across all squadrons shows that MPT expenses represent the bulk of the Air Force's operating budget. As seen in the figures, there is a great opportunity to reduce ownership costs by driving new system solutions to include human-centered costs.

With such potential savings, why aren't predecessor system costs used more within the acquisition community? The answer is that costing of new systems is normally a function of the financial management community at an Air Force system program office (SPO). This is a self-contained organization that advises the Program Manager on costs using size and weight information for system components from the engineering community to create a prediction of system costs. Clearly, this approach does not integrate other functional areas (systems engineering and human systems integration) that can gain significant benefit from using total system costs as a variable in tradeoff analyses.

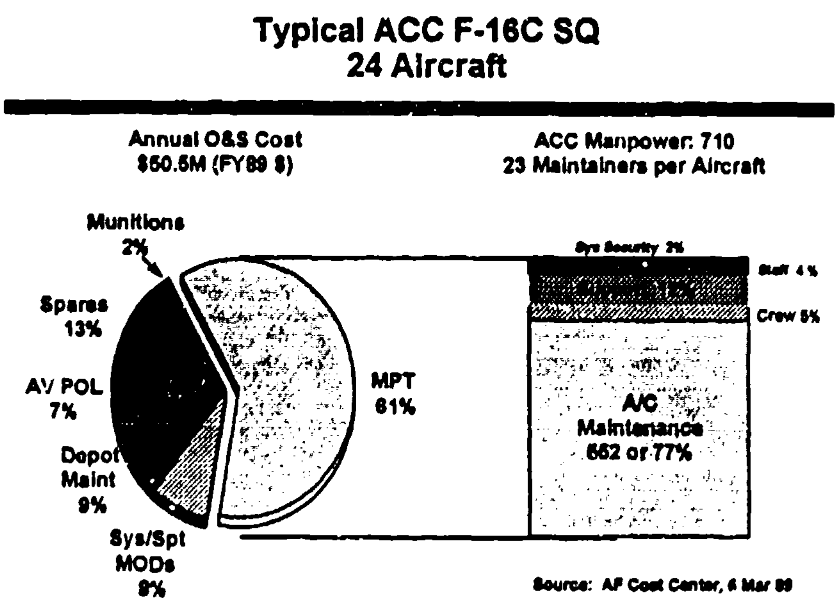


Figure 2 - F-16C Cost Summary

Historically, the Air Force has focused on the limited picture of "how much it is going to cost to procure this system" rather than the expanded view of total life-cycle cost. This short-sighted approach was practiced because of the old Air Force command structure of having separate procuring maintenance and support commands. These commands have now been merged and the "cradle to grave" management responsibility for procuring and supporting weapon systems falls squarely on the Air Force Materiel command.

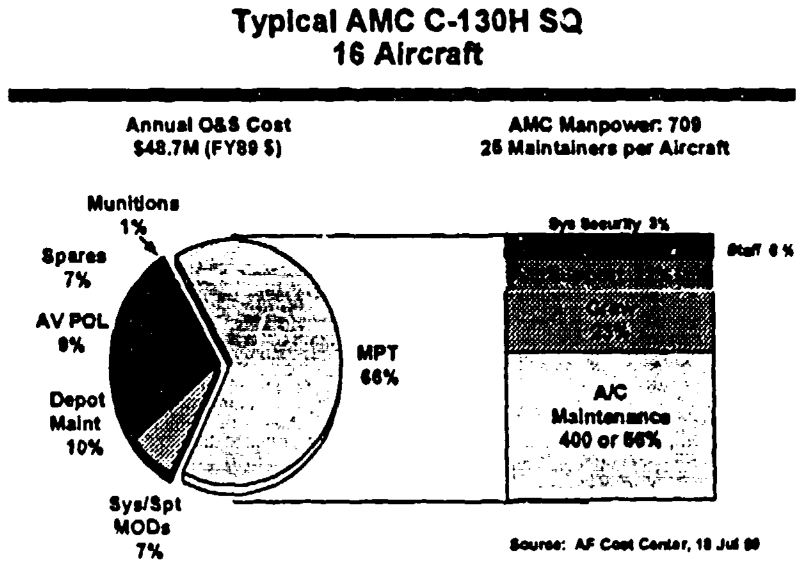


Figure 3 - C-130H Cost Summary

People costs have always been accepted as the fixed costs of doing business and were difficult for acquisition experts to quantify. This difficulty was more a problem of not knowing that such cost data were available (analysis always done within the financial management community) or not knowing how to exploit data that were available to represent these costs. From the examples given, there is clear benefit to having systems engineers and acquisition logisticians use historical cost data to help focus their development effort. SABLE is an easy-to-use cost model that is available from the Air Force Cost Center in Microsoft Excel spreadsheet file format. Some level of tradeoff analysis can be conducted through the built-in menuing and template system implemented within the model. This model provides a range of "looks" at cost data, from Air Force wide operating costs to the perspective of a single weapon system. Until more mature MPT analysis tools are institutionalized, this is one cost model that should be better integrated into the acquisition process.

MPT Research Program

The MPT research program at the Air Force's Armstrong Laboratory represents a subset of the elements contained in the DoD human systems integration program. MPT analysis represent the most challenging and least researched components within the HSI program.

MPT analysis evaluates human-in-loop costs and capabilities with intent to minimize MPT related LCCs while maximizing system capabilities. To clarify terminology, *manpower* refers to the number of positions needed to operate, maintain, and support a system in its operational environment; *personnel* to the types of people required and their characteristics and skills; and *training* to what they need to know to do their job and what resources (trainers and training systems) will be required to achieve the desired skill proficiency. "This amounts to sizing (M), describing (P), and enabling (T) the work force so that it can accomplish a given workload or function effectively and economically"[7].

To provide the best new, or modified, weapon system at the least LCC, decision makers need up-to-date data and analysis tools. The first Air Force-wide MPT conference[8] held in May 1987 identified that a major problem within the acquisition community was (and continues to be) the lack of an integrated database and analysis methodologies to effectively analyze interrelated MPT issues. The Air Force Human Resources Laboratory (AFHRL — now Armstrong Laboratory, Human Resources Directorate) launched a

comprehensive research program to meet the needs of System Program Office (SPO) decision makers in the acquisition process[9]. This program was designed to investigate data and data sources that could be used to support MPT analyses and began developing methods and tools in each of the functional domains to exploit the data. The objective was to develop a collection of methods and prototype tools that could eventually be integrated into a single integrated decision support system. The integrated system would enable acquisition and operational analysts to demonstrate the MPT related costs associated with various proposed weapon system designs, thus allowing design tradeoffs to reduce life-cycle costs.

The research culminated in the MPT Integration Branch awarding a four year, multimillion dollar advanced research and development (R&D) contract to provide acquisition decision makers with just such a DSS[10]. The contract to develop a Prototype Manpower, Personnel, and Training (MPT) in Acquisition Decision Support System (DSS) was awarded (Feb 92) to Dynamics Research Corporation and their subcontractors; Micro Analysis and Design, Rishi Technologies, and Organizational Research and Development. These companies have researchers who are intimately familiar with both the acquisition process and MPT issues and possess the needed knowledge and skilled staff capability to successfully develop an integrated system.

MPT in Acquisition DSS

This Advanced Technology Transition Development program will provide the first *integrated* tool for addressing MPT requirements during system acquisition and design. The MPT DSS is a micro-level (job task level) tool that will help analysts build a credible baseline of measurable MPT goals and constraints, provide MPT inputs needed for system tradeoff studies, allow analysts to study design alternative implications, and verify whether the completed system development achieved the MPT goals and constraints. The system will automate the extraction of historical MPT data from Air Force data bases and new system data from the Logistics Support Analysis Record (LSAR). This historical data will be used to create a baseline comparison system. As new system information is received, a notional or proposed system configuration will emerge. Finally, a suite of MPT analysis methodologies and tradeoff tools applied to a baseline comparison system (BCS) and proposed system will produce key MPT products needed to support the acquisition and design process. The purpose of the MPT DSS is to reduce defense system's life-cycle costs while improving combat readiness and supportability by identifying and resolving MPT issues *early* in the acquisition of these systems.

The MPT DSS is graphically depicted in Figure 4. This illustrates the micro-level data bases and types of analysis techniques needed to provide a comprehensive, integrated tool. The MPT DSS will support all phases of the acquisition process, from requirements analysis and determination at the Air Force major commands (MAJCOMs) to design evaluation in the SPOs. Primary analysis goals are to validate that emerging weapon system designs meet MPT constraints imposed on that system and to provide personnel and training planners with information and decision processes to establish efficient training and personnel pipelines before weapon system delivery.

The MPT DSS is based on the results of a Manpower, Personnel, Training, and Safety (MPTS) factors in the system acquisition process study completed for the Human Systems Division (HSD)[11], a front end analysis of an MPT modeling architecture[12], and an evaluation of the Army's Hardware versus Manpower methodology (HARDMAN III) suite of MPT tools[13]. Continuing close coordination with Army HARDMAN experts is ensuring compatibility between the tools and avoiding duplication of effort.

The unique capability distinguishing the MPT DSS from Army and Navy HARDMAN research is that it is a cross-domain integrated system. The manpower, personnel, and training domains are interrelated. If an analyst makes changes in any one domain, these changes will most likely effect the other two domains. For example, if you reduce the number of people you plan to use to perform maintenance on a new fighter, you then have to expand the job definitions of the remaining people to cover those tasks that would have been performed by the manpower spaces that were eliminated. Once you have expanded the job descriptions for the remaining people, your training program becomes longer and more complex for both initial and recurring training. Unfortunately, the domains are managed separately, and the reality is that when changes are proposed within any one domain (e.g., the manpower community), those changes may be coordinated with the other domains (personnel and training) but there has been no mechanism available to study the long term impacts of these changes. The MPT DSS will include functional relationships within the integrated system tradeoff analysis methods to automatically reflect the horizontal cross-domain effects of making changes in one functional domain. This cross-domain capability will greatly enhance the ability to demonstrate the cost impacts associated with different policy decisions in any one single domain.

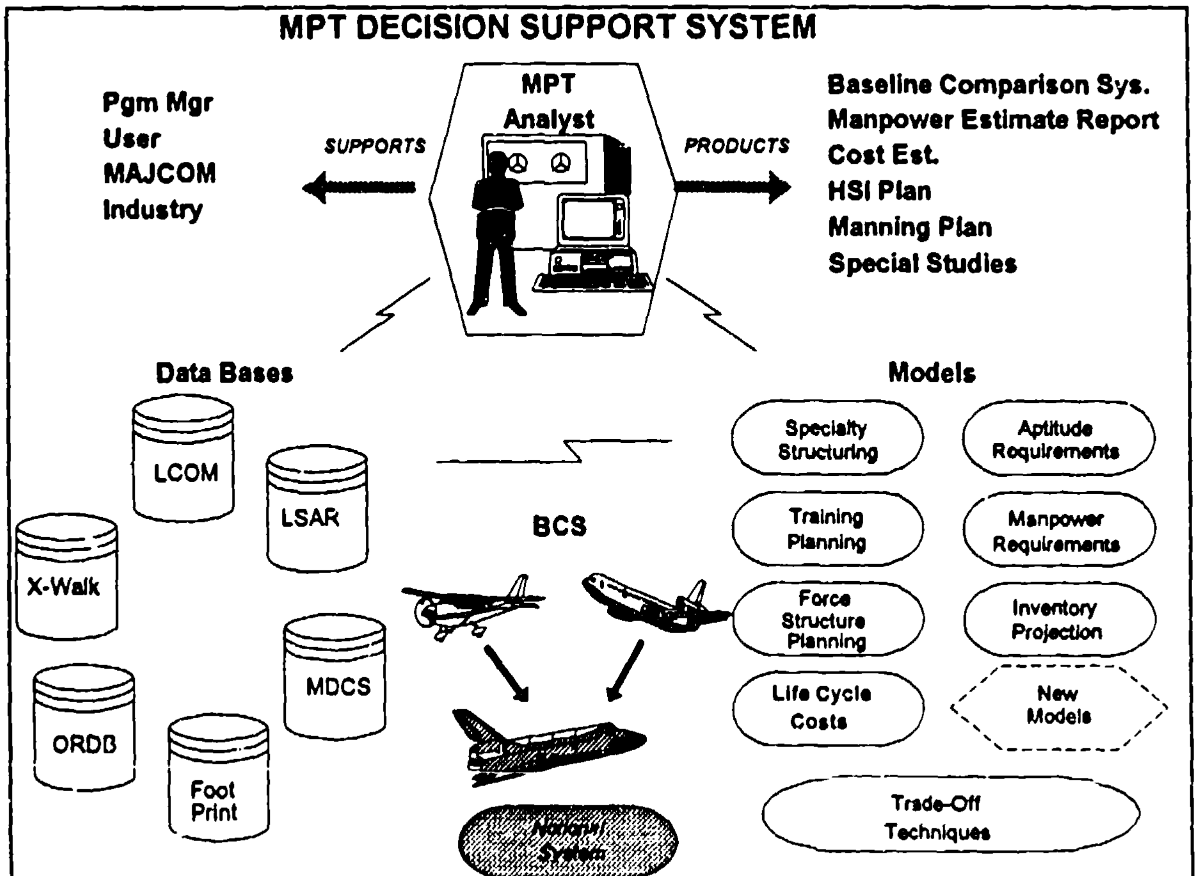


Figure 4 - Graphical View of MPT DSS

The prototype MPT DSS will focus on supporting the MPT analysis of Air Force aircraft systems but will be designed so that it can be applied to any type of system. Application to systems other than Air Force aircraft systems will require analysts to expand existing library files. The MPT DSS will concentrate on assessing MPT requirements for the maintainers and support personnel who work directly on the system in the operational units in which the weapon system will be fielded. More specifically, task-level MPT analyses will be conducted on maintainers and the support personnel whose workload is directly driven by the system. Operator crew size will be an input to the MPT DSS. Total manpower for operators, training personnel, and support personnel whose workload is not directly driven by the system will be determined by Air Force manpower standards that deal with aggregate workload, not individual tasks. The MPT DSS will contain both existing and new analytical tools.

The MPT DSS system consists of three major software components: a System Development Subsystem (SDS), a Data Base Integration Subsystem, and the Analysis Tools Subsystem. Each component is explained in the following sections.

MPT DSS Software Components

When conducting an MPT analysis, selecting the Baseline Comparison System (BCS) is a significant first step. The SDS will assist MPT analysts in constructing the BCS, populating the BCS task-level data bases with appropriate government and contractor-furnished data, and maintaining and updating the BCS data throughout the acquisition process. The SDS methodology includes techniques to match new system functional, performance, and design characteristics to those of existing Air Force equipment, at appropriate levels of system indenture.

An integrated MPT data base is needed to support the MPT DSS. The system must be capable of extracting and integrating MPT data from external Air Force data sources in a user-friendly manner. The Data Base Integration Subsystem will help Air Force MPT analysts obtain and use the input data needed for an MPT DSS application. The subsystem will request, extract, and process data from external sources; integrate input data within a comprehensive MPT DSS data architecture; and configure the data to support MPT analyses and tradeoffs.

The Analysis Tools Subsystem attempts to maximize the use of existing tools and techniques.

System Development Subsystem (SDS)

The SDS component consolidates MPT related predecessor weapon system data into a BCS. Then as design information matures, the BCS can be updated to form a proposed system description. The predecessor system is an existing system, or systems, that have components or missions similar to the new system concept. Descriptions of predecessor equipment, maintainers that repair it, manpower standards supporting it, and training courses related to it, provide a "footprint" for a new system. As identified in Logistics Support Analysis (LSA) Task 203[14], a BCS is a representative system construct composed from existing systems/subsystems (predecessor systems), support systems, and lessons learned for performing comparability analysis. The BCS components should approximate one or more of the new system functional, performance, and design requirements. As the system matures and actual design data become available through the MIL-STD-1388-2B LSA Record (LSAR), they will replace the predecessor system data. This will permit continual improvement of system design information, and provide better predictions of Air Force MPT costs and support requirements.

Comparisons between the BCS and the Proposed System are made throughout the acquisition process as the Proposed System design evolves and design alternatives are considered. Comparison of the BCS to the Proposed System requirements in the early phases of the acquisition process help identify areas of technical risk. Comparison of contractor design

alternatives to the BCS in later phases also help identify risk areas (i.e., areas for which the contractor is proposing to deliver improvements that are significantly better than what is currently being achieved) and the expected costs of reducing those risks.

Data Base Integration

This component accomplishes two tasks: it links geographically separate data sources and relates data between dissimilar databases.

One of the most difficult problems for an MPT analyst is trying to obtain all of the unrelated data from locations around the country to support the integrated analyses. This burden will be reduced by introducing a system that will automate data retrieval from geographically separate data sources. The automated system will allow the user to check a block on the user screen identifying what data are required. Then, through overnight unattended file transfer over the Defense Data Network (DDN) or by modem connection for direct attended retrievals, the data will be electronically gathered to the analyst's machine.

Another major challenge is the process of relating weapon system specific data (weapon system specific job task lists) to occupational data (e.g., which Air Force Specialties (AFSs) accomplish those tasks). In the past, this process was accomplished by gathering a group of subject matter experts (SMEs) and having them laboriously relate the data. Earlier research[15] showed that we are able to automate the process through a semantic analysis process with about an 80% text match. The SME time required is reduced by about two thirds.

This Database Integration Subsystem will provide the data needed for the BCS library and the suite of analysis tools. Maintenance, occupational, personnel, and logistic data from current systems will be used. Once predecessor systems are identified, the appropriate task-level data will be extracted. This task level data will include system costs, maintenance task data by component, occupational analysis data for job specialties working on the equipment, and training course and cost data conducted on repairing these systems and operating other support equipment. Such data bases will have their task-level data linked and extracted. If data for a specific sub-system are unavailable, then SMEs will be used. Part of this effort will require precise definitions of tasks and comparisons between the actual task statements.

Information about how to investigate the data that are available from various sources will be provided to the MPT analyst in the form of help screens. The generic content and structure of data within each source will be described. For data sources hosted at a single, or a few geographic locations, the help screens will include contact points to whom data inquiries or requests may be directed.

Analysis Tools Subsystem

This component is the core of the MPT DSS. Once the data are available, it must be analyzed and examined. The integrated set of analysis tools will be designed to support a step-wise process model for forecasting requirements based on best information available. This subsystem includes seven analysis methodologies, two tradeoff techniques, and two analysis aids, and a planning aid.

The analysis tools include a Specialty Structuring Tool to structure jobs from the ground up, at the task or task cluster level or restructure a specialty starting from an existing definition; a Personnel Aptitude and Characteristics model to ensure that the collection of job tasks does not require unreasonably high aptitude levels or physical profile characteristics that can't be supported by the current or future Air Force population; a Training Resources and Requirements Tool to project an estimate of resources needed to establish and maintain the training pipeline; Manpower Estimates Tool to determine the number of people required to operate, maintain, support, and train a single unit or squadron; a Force Structuring Tool to aggregate the manpower estimates into wings, groups, MAJCOMS, and force level

projections using approved overhead and support ratios (sufficient to support manpower estimate reports required by DoD); an Inventory Projection/Civilian Availability Model to determine whether the civilian populace can support the level of aptitude in the numbers identified from the force projection model; finally, the last model is a LCC model that will present a bottom-line dollar figure to show the MPT related costs of the system.

The individual methodologies briefly described above are useful, but more important are the techniques to permit interaction among the individual tools to tradeoff the manpower, personnel, and training domains. There is a great deal of interaction between each of the domains. Significant changes cannot be made to any one functional domain without affecting the other two. Therefore, functional relationships will be included which describe, in analytic terms, the relations between the various M and P and T factors. MPT measures of effectiveness (MOEs) will be used by the tradeoff process to provide objective criteria - identifiable and explicit- for evaluating MPT impacts of design, operation, and support alternatives. MPT control variables (i.e., the variables that the MPT community controls, and can change, to accommodate a new system) will be identified for each MPT MOE. In conducting tradeoffs, the control variables can be viewed as input variables and the MPT MOEs can be viewed as the outcome variables that are used to assess MPT impacts for all types of tradeoffs (design, support, operations). An MPT Analysis Tradeoff Aid will identify BCS or Proposed System high (cost) drivers from the MOEs. Using these high drivers, an analyst can begin conducting a sensitivity analysis by adjusting the control variables contributing to the MOE identified as a high driver. The MPT Analysis Aid will empower the analyst to conduct tradeoffs in an accelerated mode where only one variable has been changed and the entire analysis (including manpower simulation) will be rerun, or in detailed mode where the analyst can make multiple changes in several different tools. The second tradeoff technique is a Comparison Tool that will let an analyst display side-by-side results of different completed studies. The Comparison Tool uses summary reports and graphics to compare differences between the system, type, and versions. The *system* refers to the type of proposed system you are analyzing, *type* refers to the type of analysis you are conducting, and *version* refers to the specific analysis conducted. By varying the control parameters you create multiple versions of the analysis. The comparison tool allows you to compare these individual versions. These comparisons can demonstrate the relative value of different MPT approaches allowing policy options or system design differences to be studied. An analyst can converge on an optimal solution before beginning the full documentation needed to complete a study in support of required acquisition documentation.

The analysis aids are tools that will improve the IMPACTS analyst's ability to use the DSS. An integral Navigation Aid assists the user in correctly using the integrated analysis methodology for different types of studies. This technique consists of both a navigation aid visually depicting the steps necessary to complete a particular type of analysis and an extensive context-sensitive help component providing detailed topic-related assistance throughout all stages of the analysis.

The planning aid, the MPT Pipeline Tool, will assist Air Force analysts in scheduling the MPT resources associated with deploying new systems. The pipeline tool will consider training, organizational, and support pipelines, to ensure plans are developed to have trained people where and when they are needed to achieve full operational capability. Outputs from this tool include a master milestone chart that will be a PERT/CPM chart illustrating the time phasing of key MPT resourcing events based on the proposed acquisition strategy. The tool will also provide system training plan information and a forecast of required PCS.

Training Resources and Requirements (TRR) Tool - The TRR will introduce training as an acquisition variable earlier in the process than ever before.

Through the use of comparability analysis, training courses and resources associated with specialty training or specific task training will be identified. Since the BCS is the best representation of what the new system will look like made from today's technologies, an empirical data set containing course outlines, costs, and other resources associated with the existing technologies will form a training baseline. Adjustments to the baseline in terms of tasks that are trained, instructional settings used, and task training times can then reflect the cost of these changes and allow training tradeoffs to be conducted.

The objective of the TRR is to implement the front end of the Instructional Systems Development process to the degree necessary to identify and project the training resources (people, methods, aids, etc.) and training requirements (new training) for a new system or technology. The tool will be able to exploit job analysis conducted by the Air Force's Occupational Measurement Squadron or from the MPT DSS personnel aptitude and characteristics model. Based on this job analysis, the TRR will then assist the analyst in selecting tasks to be trained, assign tasks to instructional settings, determine task training times, and determine training resource requirements. The tool is intended to be used throughout a system's life cycle and includes the ability to resource training requirements for all types of Air Force training (e.g., technical training, on-the-job training, field training, etc.). Each of the major steps in the TRR process model are explained below.

Task Selection is an optional step in the analysis but expands on the Air Force's current capability. The task selection model provides capability to select and apply existing, modified, or user-defined task selection models. The TRR includes three existing task selection models: training emphasis used with occupational survey data, training recommendations provided through Logistics Support Analysis Record data, and a 3-Factor model that consist of factors that help determine the importance of training a specific task. The 3-Factor model can be better viewed as a multi-factor model and can be modified in many ways. The three principle factors within the model are: percent members performing a task, the task difficulty, and the mean operational units (e.g., flying hours, miles) between failure occurring. Additional factors, such as hazardous maintenance procedure or task criticality, can be added to the 3-Factor model based on available data and training emphasis. The model can then be a 4, 5, ... up to a 9-Factor model. In addition, the user can define up to two additional unique factors but must manually load supporting data. Each of the factors have a value range and may be further modified with a cutoff value criteria that identifies the tolerance before the model will recommend a task for training. As an aid to model selection or development, the TRR is capable of assessing availability of task data. The custom model can then be tailored to use only those factors that have data available in the system.

A significant advantage of selecting tasks for training early in acquisition is that a method will be available to validate logistics support analysis (LSA) recommendations of what tasks to train. When LSA training recommendations are received, they can be loaded into a notional system construct (database separate from the baseline comparison system) and differences between what was thought needed to be trained and what the contractor recommended for training can be identified. When significant differences exist, further analysis can be done. Through this type of process review, better training programs can be developed.

Another optional step in the TRR model is to assign tasks to instructional settings. This model needs to be used only if the analyst doesn't know the best instructional setting assignment and is looking for recommendations. Once this path of analysis is selected, the analysts can either make a direct instructional setting assignment or opt for the "setting selection model." The setting selection model is based on the training decision logic table in Air Training Command Regulation 52-22, Occupational Analysis

Program, and is the current method used for selecting tasks for training.

The TRR will also identify an occupation's training requirements and provide a graphical output depicting a training career path for an air force specialty. Each course depicted in the pipeline display is costed. This training pipeline is developed from the new system training plan and a review of existing associated courses. The TRR will then allow an analyst to build a course outline that is used as a course-level resource summary in subsequent analyses. These course outlines are built from existing specialty training standards and plans of instruction. As the Air Force's Base Training System and Advanced Training System programs are institutionalized, this development will try to extract this information in digital format. The outlines identify course modules (blocks, units, lessons) and the associated training resources (methods, time, instructors). The analyst will develop course outlines only as needed to determine the cost of a new course. If these costs are already determined because training is provided through contractor supplied training, then course outlines are not necessary.

Once the course outline is built, another model option is to determine task training times. This step predicts task training time for formal resident courses and on-the-job training. The model predicts time to train as a function of weapon system design-related task characteristics and personnel factors. This approach uses modified functional relationships first developed at the training systems program office (Aeronautical Systems Center) in its Training Analysis Support Computer System (TASCS) model. Task training times can be adjusted to reflect skill, knowledge, and ability similarity.

Finally, the model determines student inputs from the output of the MPT DSS manpower estimating tool, determines training man-weeks, and outputs instructor requirements to support the training program. This output is then sent on to the MPT DSS force structuring tool to be summarized in the Manpower Estimate Report.

The empirical baseline built during this training analysis is conceptually completed during milestone 0, concept exploration. The institutionalization of the process will inject training concerns into the acquisition process far earlier than ever before possible. As new personal maintenance aids are introduced and adapted as training aids, we will begin to see a new method of training that can be a driving force for developing maintenance and training aids of the future. This can only be done through effective analyses conducted *early* in the acquisition process.

Key MPT-related Acquisition Products and Processes

Products of the MPT DSS include the Manpower Estimate Report; Comparative Analysis (LSA task 203); support for LSA tasks 303 – Evaluation of Alternatives and Tradeoff Analysis, 401 – Task Analysis, and 402 – Early Fielding Analysis. The MPT portions of a Cost and Operational Effectiveness Analysis (COEA) can be prepared or validated. Finally, analysis summaries and Human Systems Integration plans in a format needed for the IMPACTS plan and Integrated Program Summary

Hardware and Software

The system will operate on an 80486 class (or better) microcomputer in a Microsoft Windows environment. Anticipated hardware requirements include 600 MB or more of data storage and 16 MB of RAM. The software will be object oriented and is written in the C++ programming language. There will be full documentation of the entire system, including user's manuals, design documents, technical reports, and detailed system and software specifications.

Summary

The integrated analysis tool and decision support system will assist acquisition managers, analysts, and MAJCOM planners to effectively integrate people issues (numbers, characteristics, proficiency) with equipment

(aircraft) early in the acquisition cycle. Acquisition specialists can use the structured analysis approach provided by the MPT DSS to ensure that system people costs are affordable, jobs are properly structured, and people are trained prior to the system becoming operational. The analysis methodology includes cross-domain effects (interaction between manpower, personnel, and training domains) of different weapon system designs, logistics concepts, or occupational and organizational structures. Different policy decisions can be modeled and the cost impact of those decisions identified. It will provide, in one integrated system, a means of accessing task-level data, analyzing it, and presenting it for review and study at any milestone Through use of this decision support system, the Air Force's IMPACTS program has enormous cost reduction potential in the acquisition and operational communities.

References

[1]Yates, R.W. (1989). The USAF IMPACTS Program. Air Force Principal Deputy, Assistant Secretary (Acquisition) Letter, dtd 7 April 1989.

[2]Oliveri, F. (1992). Aerospace World: GAO Hits Night Attack Plans & Airlift Questions Raised. Air Force Magazine. Volume 75, No. 8, Aug 1992., 18-19.

[3]Cole, P.A. (1991). The Impact of Manpower, Personnel, and Training (MPT) on Life Cycle Cost. Proceedings of the IEEE 1991 National Aerospace and Electronics Conference (NAECON 1991). Volume 2, 842-848. IEEE Catalog No. 91CH3007-2.

[4]Pearce, M. (1992). New Directions in Human Systems Integration. Office of the Secretary of Defense, Force Management and Personnel, Washinton D.C..

[5]Department of Defense Instruction 5000.2, Defense Acquisition Management Policies and Procedures. Part 6 Section H, Human Factors, 6-H-1 - 6-H-3. Part 7 Section B, Human Systems Integration, 7-B-1 - 7-B-5. 23 Feb 91.

[6]Gentner, F.C. (1992). Early MPTS Analysis: Using Occupational Survey Data for Comparable Systems. Wright-Patterson AFB, OH. Aeronautical Systems Division, MPT Directorate.

[7]Boyle, E. (1992). MPT Briefly. IMPACTS Bulletin. HQ USAF/MOR, Washington, DC: Vol IV, No. 1, Jan-Feb 92, 1-3, 8-10.

[8]Longmire, K.M., & Menchaca, J., Jr. (Eds.) (1987). Proceedings of the AFHRL MPT Conference. Brooks AFB, TX: Air Force Human Resources Laboratory.

[9]Bergquist, T.M. (1991). Air Force Manpower, Personnel, and Training (MPT) in Systems Acquisition Research Program. (AL-TP-1991-0021). Brooks AFB, TX: Manpower and Personnel Division, Human Resources Directorate, Armstrong Laboratory.

[10]Bergquist, T.M. (1990). Development of an MPT Decision Support System (DSS). IMPACTS Bulletin. HQ USAF/PRQE, Washington, DC: Vol III, No. 6, Nov-Dec 90, 1-3.

[11]Rosmeissl, P., Akman, A., Kerchner, R., Faucheux, G., Wright, E., Shields, J., & Waldrop, G. (1990). Analysis of Manpower, Personnel, Training, and Safety During the Acquisition of Air Force Systems: Requirements and Capabilities (three volumes) (HSD-TP-90-022). Brooks AFB, TX: Deputy for Development Planning, Human Systems Division.

[12]Rue, R.C, Black, M., Rose, D., & Bartenwerfer, D. (1991 March). Air Force Conceptual Weapon System Optimization Model (AL-TR-91-573-0202). Brooks AFB, TX: Manpower and Personnel Division, Human Resources Directorate, Armstrong Laboratory.

[13]Kaplan, J.D. (1991 January). "Synthesizing the Effects of Manpower, Personnel, Training, and Human Engineering" in Boyle, E., Easterly, J., Ianni, J., Karma, M., & Harper, S. (Eds.) Human-Centered Technology for Maintainability: Workshop Proceedings. Wright-

Patterson AFB, OH: Logistics and Human Factors Division, Human Resources Directorate, Armstrong Laboratory, 275-285.

14Air Force Logis'ics Command Pamphlet 800-17, Air Force Logistics Support Analysis Primer. 20 May 1988, 13.

15Driskill, W.E., Weissmuller, J., Hageman, D. (1992, in press). Initial Procedures for Providing Weapon System-Specific Data. Draft Technical Publication, Brooks AFB, TX: Manpower and Personnel Research Division, Human Resources Directorate, Armstrong Laboratory.

SEARCH FOR TRAINING AND HSI TECHNOLOGIES: ANALYSIS OF DoD LIVEWARE SURVEY

Frank C. Gentner and David E. Kancler
Crew System Ergonomics Information Analysis Center (CSERIAC)
Wright-Patterson Air Force Base, Ohio 45433-7022

Dr. Mona J. Crissey
United States Army Research Laboratory, Human Research and Engineering Directorate
Simulation, Training, and Instrumentation Command (ARL-HRED-STRICOM)
Orlando, Florida 32826-3276

ABSTRACT

How can the United States (US) military achieve more with the human resources it will have after completing the current downsizing efforts? By improving training effectiveness and Human Systems Integration (HSI), the DoD can leverage the people it has. To achieve this goal, the DoD has mandated a series of HSI analyses throughout the defense acquisition process. Now Government and contractor employees alike must find training and HSI technologies that help achieve better consideration of human issues during acquisition and better integration of the human into each defense system developed or modified. Recently, there has been an explosion of affordable HSI and training technologies. Despite this new emphasis, it is very difficult to identify the most appropriate technology for training development and HSI analyses. Defense acquisition managers, their contractors, and the HSI research and development community need a database of information about HSI and training tools, databases, and test facilities. They need help in identifying the technology already available in each of the Liveware domains of Manpower, Personnel, Training, (MPT) Safety, Health Hazard Prevention, and Human Factors Engineering (HFE). However, no comprehensive catalog of HSI and training technology exists. Under the sponsorship of the Office of the Assistant Secretary of Defense (Force Management and Personnel) HSI office and North Atlantic Treaty Organization (NATO) Research Study Group.21 (RSG.21), ARL-HRED-STRICOM and CSERIAC have surveyed the HSI and training communities to obtain a comprehensive database of HSI and training technologies. This paper presents highlights of the resulting Liveware database, and discusses Liveware survey collection methods, findings, and implications of this landmark survey. More than 500 HSI and training technologies have been catalogued in the Liveware database. Special emphasis will be placed on technologies critical to maintaining US military superiority while reducing manpower and training costs.

ABOUT THE AUTHORS

FRANK C. GENTNER is the senior technical analyst for the Liveware survey at CSERIAC. During his 23-year military career, Lt Col Gentner served with the initial cadre of the Air Force (AF) model MPT organization, ASD/ALH. As Chief, Analysis Division, he was responsible for developing the analytic staff and techniques used by the MPT Directorate in support of Aeronautical Systems Division (ASD) system program offices. During his AF career he also led AF and OASD activities in training requirements, development, and evaluation functions. He holds a Bachelor of Psychology and Masters in Rehabilitation from the University of Florida.

DR. MONA J. CRISSEY is the Liveware Project Manager at ARL-HRED-STRICOM. While at the Defense Training and Performance Data Center, she was responsible for design and development of the Liveware survey and database, and management of the Footprint database. She holds degrees in Education from the University of Alabama (EdD), University of Kentucky (MA) and SUNY Cortland (BS). She was formerly Manager of Administration and Program Manager for the Saudi Naval Expansion Program at Northrop Services, Inc.

DAVID E. KANCLER is a CSERIAC technical analyst assisting in the Liveware survey. Before his employment at CSERIAC, he worked on the CREW CHIEF human-model project at Wright-Patterson AFB. He recently received his MA in psychology from the University of Dayton, specializing in human factors and experimental design. He holds a BA in psychology from Ohio University.

SEARCH FOR TRAINING AND HSI TECHNOLOGIES: ANALYSIS OF DoD LIVEWARE SURVEY

Frank C. Gentner
David E. Kancler
Crew System Ergonomics Information Analysis Center (CSERIAC)
Wright-Patterson Air Force Base, Ohio

Dr. Mona J. Crissey
US Army Research Laboratory, Human Research and Engineering Directorate, STRICOM (ARL-HRED-STRICOM)
Orlando, Florida

BACKGROUND

Importance of Effective Training and HSI

The US and NATO militaries are downsizing during the post-Cold War era, while other nations are taking advantage of inexpensive military hardware and brainpower to expand their militaries. Economic pressures to spend less on military preparedness increase while tyrants and ethnic conflicts increase. Meanwhile, our training and weapon systems need to be refocused on the kinds of wars likely in the future. We are all caught up in a period of dramatic change in which it is easy to become disoriented. As acquisition people, we need to focus on how to be prepared with fewer people. To improve the people-related cost-effectiveness equation, we must find better ways to include HSI issues and technologies in the Defense materiel acquisition process, and we must leverage our investment in our people by increasing their training quality.

DoD Directives Contain HSI Requirements

Recognizing the need for more effective human-materiel interrelationships, the DoD defense systems acquisition instruction (DoDI 5000.2) documents a series of HSI analyses and data requirements to be analyzed and furnished to the Defense Acquisition Board throughout the acquisition process. Pressures to accomplish more with smaller defense forces, and the widening interest and direction in HSI as a means to this end, have accelerated the need for comprehensive information about available HSI tools, databases, techniques, and test facilities. Department of Defense Instruction (DoDI) 5000.2, *Defense Acquisition Management Policies and Procedures*, requires the "effective integration of human considerations in the design effort to improve total system performance and reduce life-cycle cost." Objectives for the human components of a system are to be established at Milestone I, and addressed, refined, and updated throughout acquisition. This DoDI enumerates appropriate HSI studies, analyses, plans, and milestone issues to be addressed at each phase of the Defense acquisition process.

Need for Available HSI and Training Technologies

To meet the challenge of these changing times and the requirements of directives, Defense acquisition personnel and their contractors need to have a set of proven HSI technologies readily available to use during each phase of acquisition. Systems development personnel need tools, data, and methods for determining HSI impacts and influencing the design process for increased human efficiency, safety, and to minimize hazards. These HSI technologies are so critical to the US technological lead that the Director of Defense Research and Engineering (1992, July) listed "human-system interfaces" and "design automation" which includes representation of people-related issues, and "environmental effects" as three of the top 11 DoD key technologies. The goal of these technologies is to help US fighting personnel perform more effectively under stressful conditions. Our people must be prepared to do more with fewer people, while remaining more protected from "harm's way". HSI and training technologies exist for just that reason, yet sometimes there seems to be a disconnect between the developer of HSI technology and the potential user. For this reason, the OASD(FM&P) HSI Office commissioned the DoD Liveware survey. The goal is to take stock of HSI technologies and index them for easy access.

NATO RSG.21

Across NATO, other nations are awakening to similar needs for easy access to HSI technology. Member countries are finding that HSI processes

can be effective in improved development/ modification of defense systems. NATO Defense Research Group Panel 8, *Defense Applications of Human and Bio-medical Sciences,* established Research Study Group 21 (RSG.21). This group, designated ***Liveware Integration in Weapon System Acquisition***, was chartered to study how the human-machine interface was addressed and how these issues were resolved during acquisition. Participant nations are listed in Figure 1. RSG.21 is chaired by Mr. Michael Pearce of the Office of the Assistant Secretary of Defense (OASD), Force Management and Personnel (FM&P)/Requirements and Resources (R&R), Total Force Requirements (TFR), HSI office.

NATO RSG.21 PARTICIPANTS

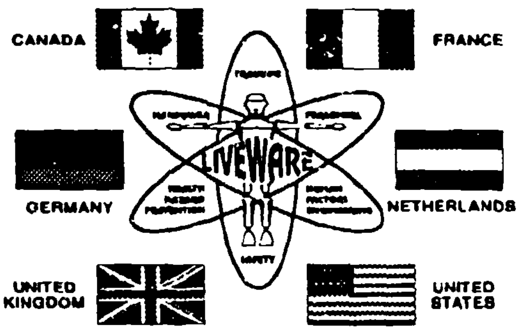


Figure 1. Liveware Participant Nations and Liveware Symbol

Liveware Defined. As RSG.21 wrestled with the difficulty in communicating concepts like Army MANPRINT, Navy HARDMAN, and AF IMPACTS -- acronyms for programs that implement HSI in the respective Services -- they coined a new term, "Liveware." ***Liveware*** collectively describes all acquisition disciplines that directly affect humans in defense systems. Liveware domains include MPT, Safety, Health Hazard Prevention, and HFE, the same disciplines involved in the DoDI 5000.2 definition of HSI. Figure 1 also displays the logo that symbolizes the Liveware concept of six domains integrated in an atom structure, with the human in the center. A bio-mechanical humanoid mannequin symbolizes the computer-aided design technology which allows integration of human issues into the design engineer's workplace, midst the creative process.

Tasking. RSG.21 was tasked to (1) identify, define, and describe the tools, techniques, and databases that enhance early consideration and integration of HSI issues into the total system; (2) evaluate these findings; and (3) identify gaps and voids for future research and development (R&D) efforts. Moving to meet this NATO-wide need, the OASD(FM&P) HSI Office tasked the Defense Training and Performance Data Center (TPDC) to develop a comprehensive database of "Liveware" information.

Project Overview and Implementers

TPDC. To build the Liveware database, TPDC developed the survey instrument to collect essential information from HSI technology owner/developers, users, and distributors. Liveware survey questions were reviewed by RSG.21 members. Since this involved translating across Service, nation, language, and scientific discipline, developing a consensus was no small challenge. Each country was to survey its own HSI community and share results with TPDC, for input into the Liveware database.

ARL-HRED-STRICOM. Shortly after initiating the Liveware survey, TPDC was disestablished and the responsibility for data collection and input was moved to ARL-HRED-STRICOM.

CSERIAC. After the survey instrument was finalized, CSERIAC's assistance was obtained as subject matter experts in the area of Human Factors, Human System Integration, and survey analysis. CSERIAC helped identify prospective technologies and Points of Contact (POCs) from literature searches and their expert network.

Liveware Project Goals

The primary goal of the Liveware survey is to be the most comprehensive study of HSI technology yet accomplished. It is to document tools, databases, methods, and facilities in all Liveware domains. The Liveware database will be available on-line and on diskette to the Government and Industry acquisition communities. This database will support effective use of HSI tools and databases throughout the acquisition process. In addition, it will store the results provided by other NATO nations. Liveware database analyses will help identify HSI technology gaps and set the research agenda to improve these technologies. The overall objective is to help DoD and NATO acquisition personnel and their contractors identify and use HSI technologies. By making HSI technologies easier to locate, we hope that they will more likely be used in producing the most cost-effective defense systems possible.

Previous Studies and Background Searches

Earlier Studies. CSERIAC performed a background/literature search to determine the ex-

tent to which HSI technology had been studied before. Nine studies were identified that covered parts of the Liveware domains (see Gentner & Crissey 1992, May) . In discussions with study authors, they identified these factors as limiting the scope of their studies: (1) the study intended to cover only one or a few domains, or one service component; (2) the breadth of the study was limited by funding or expertise; (3) participation from all domains and Services was not forthcoming and/or time was not available to personally encourage developers to submit input; and (4) the organizational infrastructure and technology did not exist within the Services during the study's timeframe (especially in the case of health hazards prevention).

Difficulties Locating Needed Technology. Comprehensive HSI technology review and comparisons are rare in the Literature. In addition, it is difficult to find POCs for HSI existing technology from literature searches. While some technical references do exist to many of these technologies, Liveware technologies are not easily located in existing technical reference databases. Often multiple cross-references are needed to find one single technology that can serve a specific need, even if the searcher is knowledgeable of the technological jargon. This dearth of easily-accessible information about HSI technologies reinforces the need for a "living" Liveware database. One could easily spend hours finding an appropriate HSI technology, just to learn that it was never completed or is no longer maintained.

METHOD

Survey Content

Survey questions were divided into the three sections. The information available from the Liveware database is listed below:

General Program Information. Section I consists of ten major areas. *Program Identification* captures the program name, acronym, description, type of technology, country of origin, community sector, state of development, availability, accessibility, and portability. The *Purpose and Acquisition Phase* covers mission area, system area, system and force level, and acquisition phase. The next three areas cover *Hardware Requirements, Software Requirements*, and *Linkages* to other tools/databases. *Documentation* captures the names and dates of the technical reference and user instruction documents, data output mode, and availability of data field descriptions and data record layout. The *Validity* area has product validation information. The three final areas are text fields covering *Assumptions, Limitations*, and *Remarks*.

Descriptive Information. Section II identifies the Liveware domains addressed by the program, applicable categories within each domain, and environmental areas of concern to safety and health hazard programs. In addition, if the program integrates several domains, the method of integration (vertical and/or horizontal) is specified.

Owner/User Information. Section III covers multiple areas. Not only is the owning organization identified with a POC, but multiple users and their organizations can also be identified. For each POC, the following information is collected: organization name, address, and telephone number, user work discipline, domains applied, and frequency of use. For a more detailed description of the survey, see Gentner & Crissey (1992, May).

Survey Administration Strategy

Maximum publicity was sought by publishing and presenting papers/articles at technical forums and in professional publications, such as the National Aerospace and Electronics Conference (NAECON), Human Factors Society, Interservice/Industry Training Systems and Education Conference (IITSEC), DoD Human Factors Engineering Technical Group (DoD HFE TG), *CSERIAC Gateway*, and other conference/ workshop proceedings. The CSERIAC specialized literature searches identified potential technology POCs, who were sent Liveware surveys, followed-up by phone and fax. As surveys arrived, CSERIAC used "networking" techniques to identify other technologies and POCs. For those technologies with no POC participation, existing literature was used to develop a survey entry. When possible, those literature entries were coordinated with the POC.

Survey Database and Analyses

Prototype Databases. Survey results were entered in a prototype PC FOCUS database, and later into a Folio Views hypertext infobase. The database displayed matrix-type (cross-tabulation) printouts of survey variables. The infobase enabled instant word combination searches.

Survey Analyses. Survey analyses were conducted by Frank Gentner, Dave Kancler, and Dr. Mona Crissey using matrixed printouts from the Liveware database. Descriptive statistics were used to highlight the existence of technologies in various categories and to look for trends. At press time, detailed analyses of these 3 following

Table 1
TECHNOLOGIES IN LIVEWARE DATABASE BY SERVICE/INDUSTRY
(As of April 15, 1993)

LIVEWARE DOMAIN	UNITED STATES AIR FORCE	ARMY	NAVY/ MARINES	OTHER GOVT.	INDUSTRY	UNIVERSITIES	TOTAL BY DOMAIN
MANPOWER	54	44	15	26	103	7	248
PERSONNEL	43	44	12	25	99	8	231
TRAINING	68	52	37	33	126	8	324
SAFETY	27	18	8	19	87	6	165
HEALTH HAZARDS	21	16	7	18	71	4	137
HUMAN FACTORS ENGINEERING	48	42	11	30	112	11	254
INTEGRATION	38	23	15	23	79	8	186
NUMBER OF TECHNOLOGIES IN DATABASE	116	85	52	60	174	13	500

NOTE: Each technology can impact more than one domain

groups have been conducted: Total HSI Survey (579 participants, 500 technologies, and 295 users), Human Factors Engineering-related technologies (301 HFE total participants, 254 HFE-related technologies, and 137 HFE users), and Training-related technologies (378 total participants, 324 training technologies, and 198 training users). This paper will (1) examine the representativeness of the survey sample, (2) present results of the total HSI survey, then (3) concentrate on the training technology findings, comparing them with the total HSI findings.

RESULTS

Adequacy of Survey Sample

Participants Outnumber Technologies. Liveware survey participation as of April 15, 1993 totaled 579 owners, developers, users, and distributors covering 500 technologies. Since more than one user could participate for each technology, the number of participants exceeds the number of technologies in the database by 79.

Number by Domain and Service/Other. Table 1 presents a listing of technologies by Service/other organization and by Liveware domain. The Training domain has had the greatest number (324) of programs listed. The HFE and Manpower domains are next with 248 and 254 programs, respectively. The lowest numbers of technologies by domain are in the Safety and Health Hazards domains, but they still have achieved 165 and 137 "hits" respectively. Participation by DoD Service shows the AF has 116 and the Army 85 technologies, while the Navy/ Marine Corps grouping has 52 technologies listed. It is possible that the Navy is either under-represented, or that it has fewer HSI technologies than the AF and Army. When we contacted personnel from a Navy lab that specializes in MPT issues, they indicated there was no HSI-related research going on at that lab, or anywhere in the Navy to their knowledge. The showing from Industry is quite good, with 174 technologies listed. The least participation came from academia, with only 13 listed. Academia coverage could be sparse for one of these reasons: Academicians did not make the connection between their technologies and defense systems acquisition; they were not interested (and some stated so); or maybe they might not have many HSI tools. Thus, if this sample is deficient, it probably would be in Navy and university-developed technologies.

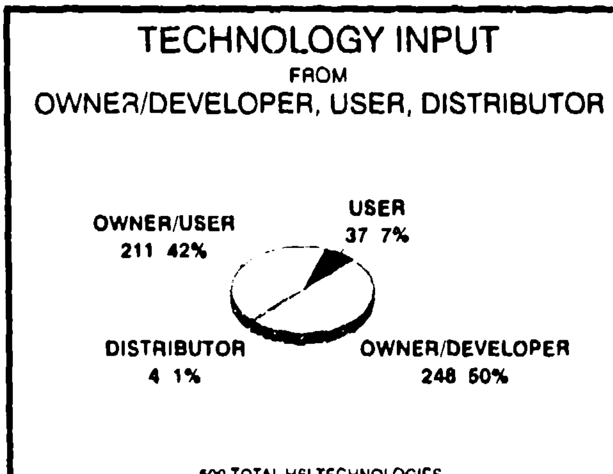


Figure 2. Technology Input Source

Technology Descriptions Come Primarily from Owner-Developer-Users. Figure 2 presents the technology input source. Over 50 percent of the input used to describe each technology came from owner/developers, 42 percent from owner/users, one percent from distributors, with less than seven percent or 37 technology inputs coming only from users (without developer/owner input). Thus, the source of the information contained in the Liveware survey appears to be authoritative, with more than 93 percent input from owner-developers, owner-users, and distributors.

Total HSI Survey Findings

The number of technologies identified as supporting each Liveware domain is displayed in Figure 3. Specific definitions of these domains are presented in last year's IITSEC Liveware paper (Crissey and Gentner, 1992, November), and are similar to those in DoDI 5000.2.

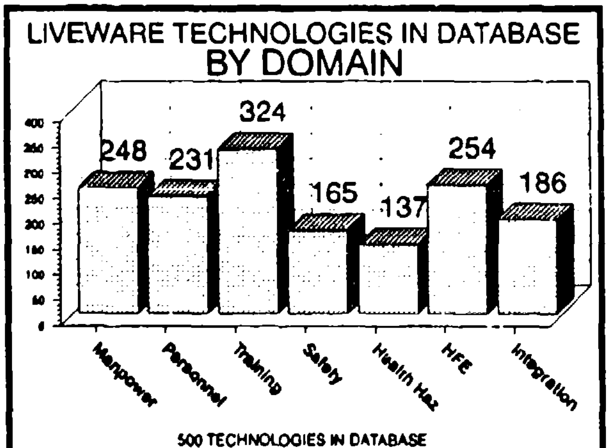


Figure 3. Liveware Technologies by Domain

Manpower. Of the 248 manpower-related technologies, 194 assisted the development of operator manpower, 181 maintenance manpower, 114 support manpower, 99 instructor manpower, with only 28 technologies supporting casualty estimates (see Table 2). The number of existing tools, databases, techniques that already exist appears quite adequate on the surface, However, by examining the technology listing, one can see that these technologies range from very specialized models good for only one class of weapon system, to ones that are so generic that to use them involves labor-intensive development of the databases and task network models to provide a manpower estimate. Some of these models simply project the number of authorizations needed to field weapon systems once the manpower per system or unit has been developed, and thus, ac-

TABLE 2

HSI Technologies by Domains & Subdomains

DOMAIN / Subdomain	NUMBER OF TECHNOLOGIES	
MANPOWER	248	
Operator		194
Maintainer		181
Support		114
Instructor Trainer		99
Casualty Estimates		28
PERSONNEL	231	
Occupational Classification		106
Selection		89
Skills, Knowledge, Ability		178
High Driver Tasks		90
TRAINING	324	
Methods/Media		132
Op Tempo		36
Effectiveness		85
Skill Decay		56
Training Resources		153
ISD		174
Special Training including: (Simulators, CBT, Embedded)		194
Instructional Systems Development	174	
Analysis		19
Development		11
Design		1
Implementation		9
Evaluation		7
Combined Steps		127
SAFETY	165	
Human Safety		154
Equipment Safety		100
Thermal (heat, cold, humidity)		60
Mechanical (shock, vibration)		106
Radiation & Directed Energy		61
Chemical Threats		64
Electrical		85
Atmospheric Pressure		51
HEALTH HAZARD PREVENTION	137	
Psychological		48
Physical		125
Thermal		52
Mechanical		93
Radiation & Directed Energy		54
Chemical Threats		62
Electrical		85
Atmospheric Pressure		46
HUMAN FACTORS ENGINEERING	254	
Mission, Function, Task Analysis		136
Task Performance & Workload		177
Human-Machine Interface		139
Information Transfer		88
Workspace & Anthropometry		86
Environment, Life Support		78
INTEGRATION	186	
Vertical (only)		35
Horizontal (only)		53
Both		84
Unspecified		14

complish only one piece of the manpower estimation job.Other technologies counted here include those that only tangentally assist with development of manpower figures and primarily belong in another domain. (For each of the Liveware domains and subdomain descriptions, see Table 2 for the number of technologies in each category.)

Personnel. While the 231 technologies purport to assist with personnel decisions, many of these are, in fact, training tools that assist with skill, knowledge, and abilities (178) and few enable the projection of the skill requirements driven by a particular design solution. Other personnel technologies assist with occupational classification (106), personnel selection (89), and identification of high driver tasks (90).

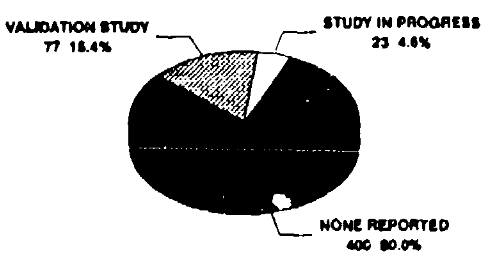
Training. Of the 324 training-related technologies, most (194) were associated with special training systems (e.g., simulators, etc.), Instructional Systems Development (ISD) (174), training resources (153), and method/media (132). Relatively few (36) were associated with OP Tempo, skill decay (56), or training effectiveness (85). Table 2 also presents the number associated with each phase of ISD. Most technologies covered multiple ISD phases.

Safety. One hundred sixty-five safety-related tools supported both human (154) and equipment (100) safety. Mechanical (106) and electrical (85) were the areas most addressed, with atmospheric pressure addressed by 51.

Health Hazards Prevention. Of the 137 Health Hazard Prevention technologies, only 48 were psychological, while 125 were concerned with physical aspects. The most supported areas were mechanical and electrical hazard prevention, and least supported was atmospheric pressure.

Human Factors Engineering. HFE enjoyed the second largest participation. Of the 254 HFE technologies, 177 were associated with performance and workload and 136 with mission, function, and task analysis. The fewest HFE technologies were associated with life support (78).

VALIDATION
NUMBER OF VALIDATION STUDIES/REFERENCES



500 TOTAL TECHNOLOGIES

Figure 4. Number of Validation Studies

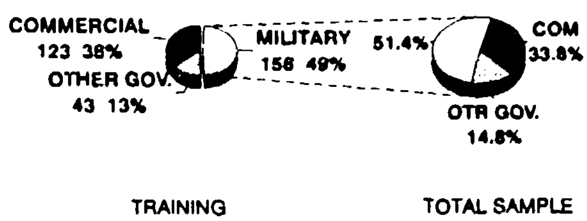
Integration. Among the most important functions of HSI tools is integration. Over 180 technologies claimed to achieve some form of integration (general category). Varying numbers addressed horizontal (35), vertical (53), or both types of integration (84). Notable is that fewer than 45 technologies integrated all domains.

Validation of Few. Of the 500 technologies, only 77 were validated and 23 had validation studies in progress for a total of 20 percent. This means (see Figure 4) that 80 percent did not have or report validation studies, a major deficiency in developing the credibility of HSI tools.

Training Findings & Comparisons to Total HSI

Technology Ownership. Most Liveware training and technologies were owned by the military (about 50% for both) and other government organizations (13-14 %), while 34-38 percent were commercial tools. Slightly more training technologies were proprietary than were other HSI tools (33 % versus 29%) leaving nearly 70 percent of technologies listed in the Liveware database as non-proprietary (see Figure 5).

TECHNOLOGY
OWNER SECTOR



324 TRAINING, 500 TOTAL TECHNOLOGIES

Figure 5. Technology Ownership

Technology Type. Seventy-one percent of training technologies were tools, compared to 62 percent of overall HSI tools. A lower percentage of training technologies was databases (17%) compared with HSI databases (25%). Techniques were about the same percentage (12%) in both areas (see Figure 6.)

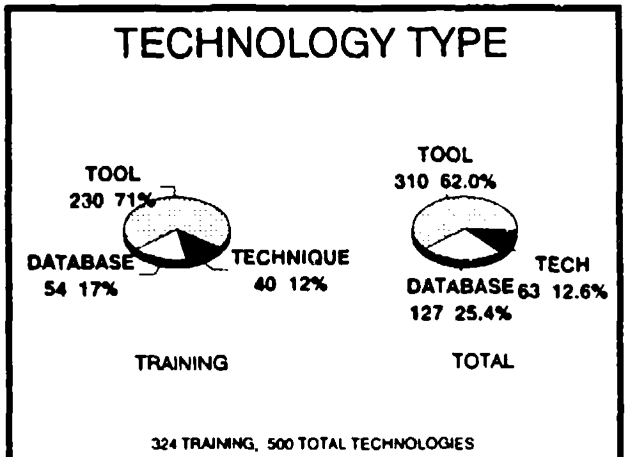


Figure 6. Technology Type

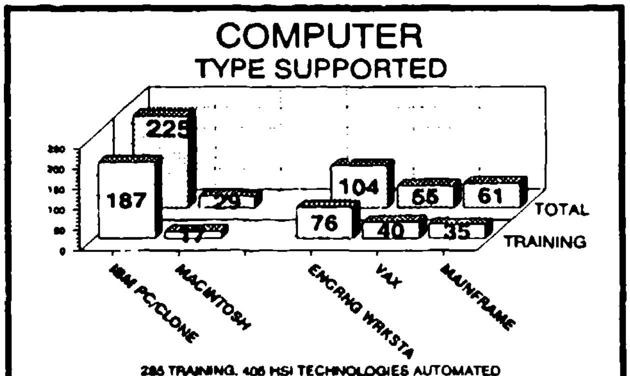


Figure 7. Comparison of Computer Types Supported

Computer Type Supported. Of 405 automated technologies, the most supported computer types were the IBM PC/clone (225) and engineering workstation (104). Only 29 technologies were identified as Macintosh-based, despite an intensive literature and expert network search for Mac-based tools. Training technologies followed suit. Most were PC-supported with about the same percentage of other computer support as the total sample, except that they had relatively fewer mainframes (see Figure 7).

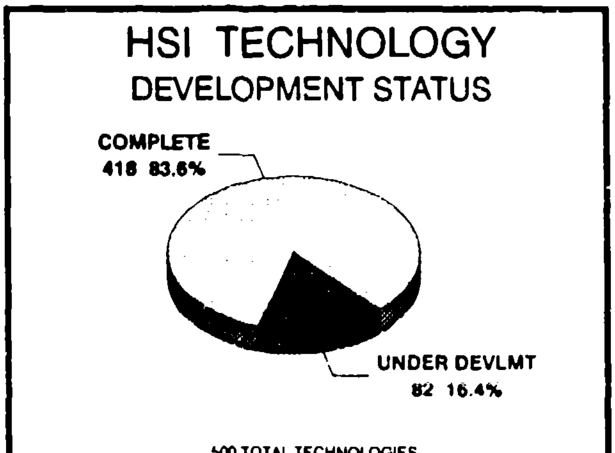


Figure 8. Percent Completed

Development Status. More than 80 percent of both training and HSI technologies listed in the Liveware database are complete and ready for use. This should quell the rumors that HSI technology is all "vaporware" (see figure 8).

System Areas Addressed. Both training and HSI tools addressed all system areas in high numbers and nearly equal proportions. Aircraft systems were the most addressed (see figure 9).

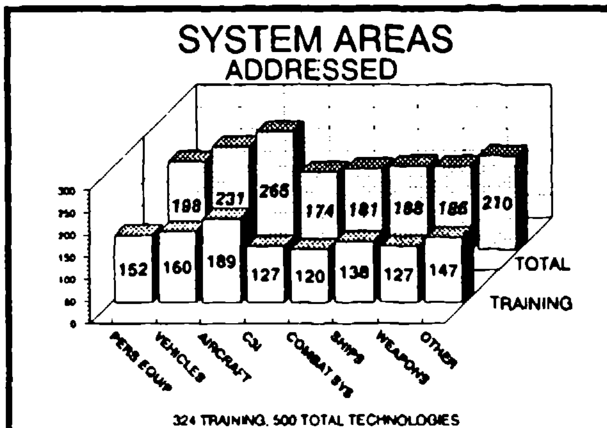


Figure 9. System Areas Addressed

System Levels Addressed. For both training and HSI technology, single systems are most addressed, with mixed systems and missions, and component level least addressed. (see Figure 10).

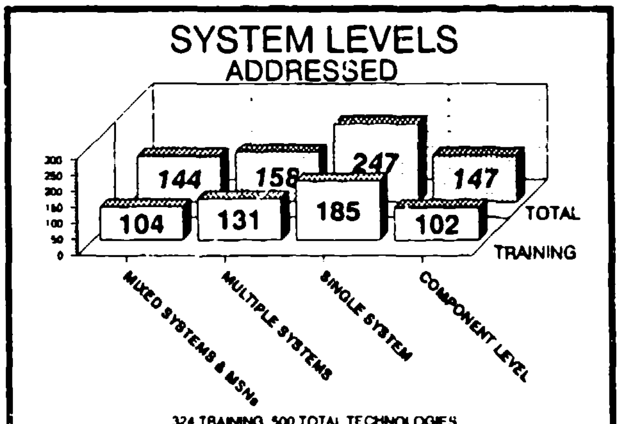
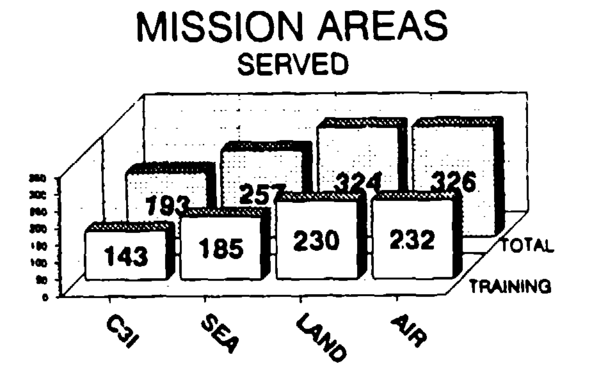


Figure 10. System Levels Addressed

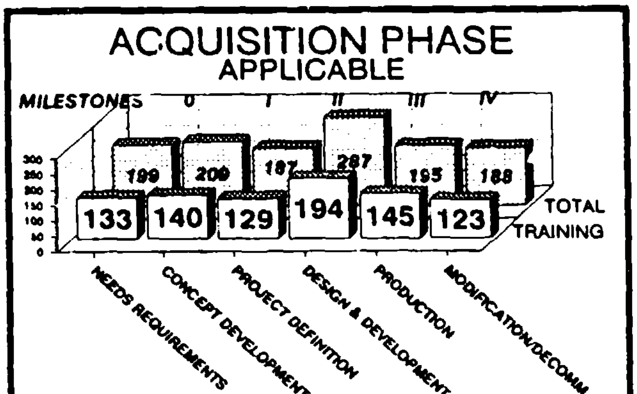
Mission Areas Served. All major mission areas were served, with greatest emphasis on air and land (as one might expect, given the lower participation by the Navy. (see Figure 11).



324 TRAINING, 500 TOTAL TECHNOLOGIES

Figure 11. Mission Areas Served

Acquisition Phase Applicable. Both training and HSI technologies supported all phases of acquisition. The most frequently supported phase was design and development (to use US terminology, Engineering and Manufacturing Development). (see Figure 12).



324 TRAINING, 500 TOTAL TECHNOLOGIES

Figure 12. Acquisition Phase Applicable

About the User

User Work Discipline. Figure 13 presents user work discipline. The largest number of HSI and training technology users in the survey was engineers and designers (60 and 31, respectively). Scientists and researchers were second, with 55 and 27, respectively. In addition, for both HSI and training-specific tools, users reported using their technologies most frequently in either "daily" or "as required" categories.

DISCUSSION

Coverage of Liveware Tools

Although a considerable number of tools that cover all domains, subdomains, system areas and levels, missions, and acquisition phases, some gaps were found in the survey data.

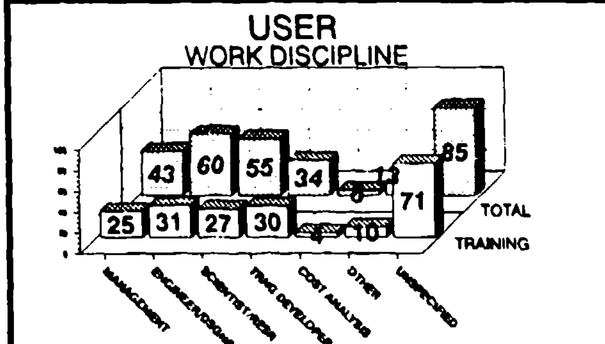


Figure 13. User Work Discipline

Integration. While 186 technologies indicate that they accomplish some form of integration, a maximum of 84 can specify the type (vertical or horizontal). Although one could argue that respondents didn't understand the question, it is more likely that the entire HSI area lacks a thorough network of integrated tools and databases. One clue to the need for integration and linkage could have been the linkage question; however, very few answered the questions (60) and their answers appeared as though the question was misunderstood. The literature is full of complaints about the number of HSI tools that have no database on which they can be run, or databases that are too expensive to build. It will take a study to show the input, process and output of these tools to determine the extent of integrated tools for HSI. The Liveware survey could be extended to accomplish this level of analysis, using the present data as a start.

Utility. The actual gaps occur in the utility of a technology and how cost-effective it is to use. The Liveware survey did not ask evaluative or cost-benefits questions about the technologies. Later versions, with special mailings to HSI technology users could help answer the question of cost-benefits, and disconnects in using this technology during acquisition.

Missing Details. All areas addressed in the survey appear to be covered except integration, the categories used in the Liveware survey were broad. For example, the Human Factors area had only six subcategories, while the CSERIAC taxonomy of human factors includes 15 major areas with thousands of subcategories. To determine whether there was adequate coverage, one would need to make a multidimensional matrix of the type of analysis by the type of system and level, Service, and by type of person (operator, maintainer, trainer, etc.) to see which

cells of the matrix were inadequately supported. Because the categories in the Liveware survey were broad, it is difficult to identify missing tools from the present dataset. If the Liveware survey were considered a first step toward identifying an optimal taxonomy of technologies for use in HSI, future versions could more carefully define each element of the taxonomy and could better characterize tools.

Missing Validity Studies. Perhaps the most significant finding of the study is the fact that 80 percent or more of the HSI technologies reported no existing or in-progress validity study. To develop and maintain credibility with program offices and the user of the technology, validity studies need to be planned and executed to demonstrate the worth of HSI technology. For those technologies that have published validity studies, the Liveware database can help find easy access to these documents, as well as differentiate those validated technologies from others.

Uses of Liveware Survey Database

Assessment Aid. Survey data, displayed and analyzed using the Liveware database, can help identify the technology available. To the extent that missing technologies can be compared with those existing in the Liveware database, deficits can be identified. This will provide a basis to marshal R&D resources. Information will be shared NATO-wide, thus making maximum use of existing technology wherever it exists. This could ultimately save HSI technology development costs.

Technology Choice Aid. The Liveware database does not rate or rank individual technologies, nor does it provide descriptive information in great detail. It does provide enough information to the analyst, program manager, or developer to narrow the list of appropriate technologies to those of value for a particular domain, task, or acquisition phase. By providing a broad range of information in an easily-queried summary format, the user can quickly narrow searches for appropriate tools. By providing POC information about tool developers and users, the pursuit of in-depth information about tools is easily accomplished.

Secondary Benefits. By making the database available and widely advertised to the acquisition community, it is likely that the Liveware program will have these positive benefits. It will (1) promote state-of-the-art information sharing; (2) help potential users of HSI technology easily find what they need; (3) encourage the use of HSI tools and databases; (4) help identify available technology and gaps; and (5) help set and substantiate the HSI research agenda.

REFERENCES

Crew System Ergonomics Information Analysis Center (CSERIAC) (1992, March). *CSERIAC Taxonomy.* Wright-Patterson AFB, Ohio: Author.

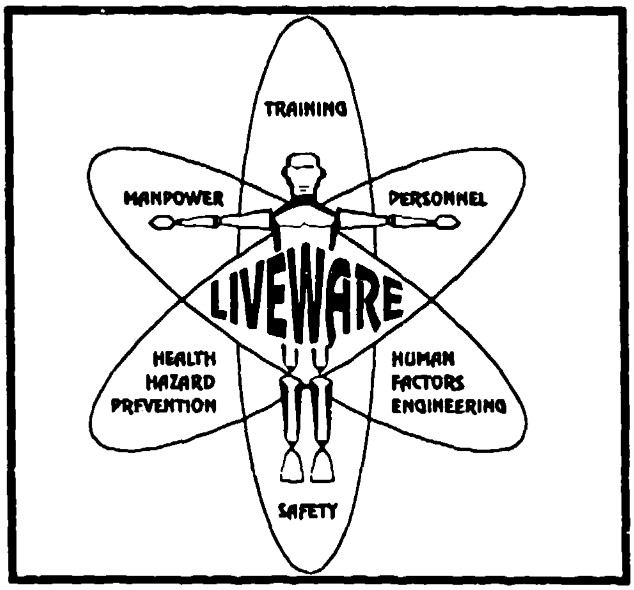
Crissey, M. J. and Gentner, F.C. (1992, November). Locating Human System Technologies for the Acquisition Process. *The 14th Interservice/Industry Training Systems and Education Conference (IITSEC 92) Proceedings*, 386-395.

Department of Defense (1991, February). *DoDI 5000.2, Defense Acquisition Management Policies and Procedures, Part 7, Section B, Human Systems Integration.* Pentagon, Washington, DC.

Department of Defense, Director of Defense Research and Engineering (1992, July). *DoD Key Technologies Plan* (AD-A253-692). Pentagon, Washington, DC.

Gentner, F.C. and Crissey, M. J. (1992, May). Liveware Survey of Human Systems Integration (HSI) Tools. *NAECON 92 Proceedings*, 2, 715-723. Dayton, OH: Institute of Electrical and Electronics Engineers.

Gentner, F.C. and Crissey, M. J. (1992, October). Liveware Survey of Human Systems Integration (HSI) Technologies: Need for Comprehensive Survey and Available Database. *Proceedings of the Human Factors Society 36th Annual Meeting*, 2, 1133-1137. Santa Monica, CA: Human Factors Society.



TAKING THE GUESSWORK OUT OF PROGRAM MANAGEMENT UTILIZING COTS SOFTWARE

Richard L.Peters
Metters Industries, Inc.
Orlando, Florida

Edward M. Beckmann
Metters Industries, Inc.
Orlando, Florida

Steven J. Johnson
Metters Industries, Inc.
Orlando, Florida

ABSTRACT

This paper will discuss the tailoring and utilization of Commercial Off-the-Shelf (COTS) software to perform Program Management. The requirements were to provide a COTS approach to Program scheduling and tracking, determining and tracking personnel resource requirements, and to depict program funding status. Specifically, this paper will address the COTS software utilized by the Simulation, Training, and Instrumentation Command (STRICOM) to perform Program Management from the receipt of a draft Operational Requirements Document to system delivery.

Typically Program Managers have not had sufficient automated tools for Program Schedule planning and tracking, personnel resource forecasting and utilization, and funding overview in an easy to use format. The ability to cross-check the deliveries specified in a RFP and Section F of a contract has been labor intensive in the past. This paper will discuss the integration and utilization of Microsoft Project and Excel to accomplish these tasks easily and in a timely manner.

In discussing the utilization of COTS software for Program Management, the paper will address the requirements for the STRICOM system, its capabilities, and the benefits received from its use. The paper will also discuss the system's applicability to other organizations, both government and defense contractor.

Automated program management for all systems, ACAT I to ACAT IV, is required to schedule and track shrinking resources and to conduct real time "what if" drills in order to make intelligent program decisions. The utilization of the STRICOM system is one method of accomplishing these tasks.

ABOUT THE AUTHORS

LTC RICHARD L. PETERS (USA, Ret.) is the Senior Principal Analyst for the development and implementation of the STRICOM Scheduling and Tracking Tool. He is a retired Lieutenant Colonel with more than 25 years of service, His last three years were served at PM-TRADE/STRICOM. He holds a B.S. in aeronautics from Embry-Riddle Aeronautical University and is a graduate of the Defense Systems Management College. He is presently the President of Advanced Military Technologies, Inc., a consulting firm in Orlando, Florida.

MR. EDWARD M. BECKMANN, PE is the Senior Program Analyst for the development and implementation of the STRICOM Scheduling and Tracking Tool. He retired from Government Service, having served at PM-TRADE as Chief, Technical Support Division and as Chief, Resources Management Division. He is a Registered Professional Engineer in Illinois and holds a B.S. in Electrical Engineering (Communications and Electronics) from the University of Iowa. He is also a graduate of the Defense Systems Management College.

MR. STEVEN J. JOHNSON is the Senior Software Engineer for the development and implementation of the STRICOM New Work Brief/Project Acceptance Committee Tool. He is a Lieutenant Commander in the U.S. Navy Reserve, and has spent the last five years supporting various projects for both STRICOM and NTSC. He holds a B.A. degree in Political Science from Vanderbilt University, and is currently pursuing a M.S. degree in Computer Science.

TAKING THE GUESSWORK OUT OF PROGRAM MANAGEMENT UTILIZING COTS SOFTWARE

Richard L. Peters
Metters Industries Inc.
Orlando, Florida

Edward M. Beckmann
Metters Industries, Inc.
Orlando, Florida

Steven J. Johnson
Metters Industries, Inc.
Orlando, Florida

INTRODUCTION

Program Management for acquisition of weapons systems and training devices within DOD is an extremely complex process. By providing tools for front end planning and scheduling of a military acquisition and the capability of tracking the completion of scheduled events, after contract award, the job of the Program Manager is made somewhat easier. Utilizing Commercial Off-the-Shelf (COTS) software as the basis for these tools reduces development time, risk, and cost normally associated with the development of a specialized software program. Two of the tools provided to the U.S. Army Simulation, Training, and Instrumentation Command (STRICOM), which utilize COTS software as their basis will be discussed. The tools are the Close Combat Tactical Trainer (CCTT) Project Management and Tracking Tool and the New Work Brief/Project Acceptance Committee (NWB/PAC) Tool. Requirements for these tools were developed by STRICOM, based upon earlier findings that defined their Local Area Network (LAN) and Management Information System (MIS).

THE REQUIREMENTS

CCTT Project Management Tool

The requirement for this tool stated that it would be a PC-based management software tool that would operate within the MS-DOS 5.0 operating system and in a Microsoft Windows environment. This COTS software based tool was to run on IBM PC/AT compatible 386/33 MHz machines and IBM PC/AT compatible Intel-386 based Compaq notebook computers. The tool would include defining, tracking, and maintaining project schedule data, Action Item Tracking, and Project Schedule Change Tracking.

The requirement also stated that the government would provide the hardware and the following COTS software: MS-DOS 5.0, Microsoft (MS) Windows, MS Excel, and Word Perfect for Windows. The contractor was to obtain and provide the following COTS software: Harvard Graphics for Windows, the latest version of PC Plus, MS Project for Windows, and other PC-based COTS software required to accomplish the effort (i.e., dBase III+, Dbase IV, or Paradox).

Utilizing the above, the effort required Schedule Creation and Schedule Maintenance for the CCTT Program. Schedule Creation consisted of an independent evaluation of the CCTT Request for Proposal (RFP) to identify deliveries, meetings, conferences, test activities, and key milestones up to and including the Milestone III decision. It also required the review of the DOD 5000 Series of instructions and directives to identify the activities, events and milestones necessary to obtain a Milestone III decision for an ASARC program. The data from both the RFP and DOD 5000 Series was to be input into MS Project to establish an initial CCTT Project Schedule. Schedule Maintenance required working with the CCTT project team to gather detailed schedule data, i.e., completion of events/tasks, to maintain a current CCTT Project Schedule. Schedule Maintenance also included updating views, scheduled start and end dates, float time, interdependency and recording actual start and finish dates and the percentage of completion of each task in the schedule.

NWB/PAC Tool

The NWB and PAC procedures were to be utilized to enhance the visibility of any new initiative likely to require the expenditure of resources, either funding or manpower and to authorize the commitment and assignment of STRICOM resources in support of new work efforts. The requirement for this tool stated that it would be a PC-based system that would operate in the same software and hardware environment described in the requirements for the CCTT Project Management Tool above. This tool was to be maintained on the STRICOM LAN and be accessible from any PC on the LAN. There were five main requirements that had to be met with this tool. First, was a requirement to provide a standard format for the STRICOM New Work and PAC briefings. This entailed providing the users a data entry interface that allowed

for the fast and simple creation of briefing material in a standard format, and providing the user with a professional looking presentation (slide show). Second was the requirement to provide a set of generic project templates from which a project director and his matrix team could build their own schedule. The generic templates were to start with the draft Operational Requirements Document (ORD) and end with system delivery. Third was a requirement to provide a method to assign man-hour requirements, by labor discipline (e.g., Project Engineer, Software Engineer, Logistics Specialist, etc.) to a scheduled task. The fourth requirement was to provide a five year project funding summary based upon the types of funding for the project and whether the amounts were funded or unfunded. The last requirement was to provide a comprehensive on-line help facility for all portions of the tool.

SELECTION AND INTEGRATION OF COTS SOFTWARE

CCTT Project Management Tool

Part of the selection of software for this tool was directed by the user, that being MS Project for Windows. This COTS software provided the ability to create the schedule for the CCTT Project, for instance: Contract Data Requirements List (CDRL) items delivery, Program Reviews and Conferences and their minutes, Program major events (i.e., SSR, PDR, CDR, PCA, etc.), and events required by the DOD 5000 Series instructions. MS Project also provides the ability to track a schedule once it is created and can provide the variance from planned start and finish to actual start and finish. Therefore, this COTS software provided the ability to meet all of the requirements except for Action Item Tracking and Schedule Change Tracking.

To meet these requirements, Superbase 4 was selected because, at the time, it was the best database program with an MS Windows interface. To accomplish the Action Items Tracking, an Action Item Form was created using Superbase 4. A Macro was developed in MS Project that provides access to the Action Item Form. A button to represent the Macro, was placed on the MS Project Tool Bar for ease of operation. The Action Item Form allows recording of an Action Item, the person who created it, a suspense date, the organization and person responsible for resolution, and when the action was completed. For the purpose of Schedule Change Tracking a Change Comment Form was created in Superbase 4, and utilizing its Dynamic Data Exchange (DDE) capability an interface with MS Project was established. The Change Comment Form can be tied to any task in MS Project and a button for it was also added to the Tool Bar, with an associated Macro. This provided the capability to record when the schedule was altered, by whom, and why. These simple actions fulfilled the software requirements for the CCTT Project Management Tool.

NWB/PAC Tool

The selection of software for this tool had the goal of quickly developing an easy to use application with a comprehensive help facility. To achieve this goal, the development team used available COTS software packages to their fullest extent. The packages selected were MS Project and Excel for Windows, since these applications were already in use at STRICOM. MS Project was selected for its scheduling capability. Its resources capability was not used because of its limitations and difficulty of use. Excel was selected for handling resources, funding and costing information and its graphics presentation capabilities. To meet the diverse requirement for this tool, the main portion of the application was coded using C++ and the Microsoft Foundation Classes (MFC) as the base development tool. The base development tool was designed and developed to control the flow of information, both within Excel and Project, and from Project to Excel.

NWB Tool - A set of eight screens were developed, utilizing C++, to standardize the New Work briefing. The screens used for creating a NWB (Figure 1) were titled as follows: Cover/Title Slide, Description, Milestone, Source of Requirement, Funding, Funding Remarks, Issues or Open Actions, and Other Information. Each of these screens contain character entry boxes and/or pull down menu selections for inputting information. Each screen was provided with the capability to move to the next or previous slide, to cancel creation of the brief at any point, and to display on-line help. The ability to edit the screens was also provided. After all data is entered and edited on the screens, the capability to run a slide show, or print a copy of the presentation, was provided. The slide show is generated from the input screens using the MS Excel slide show add-in.

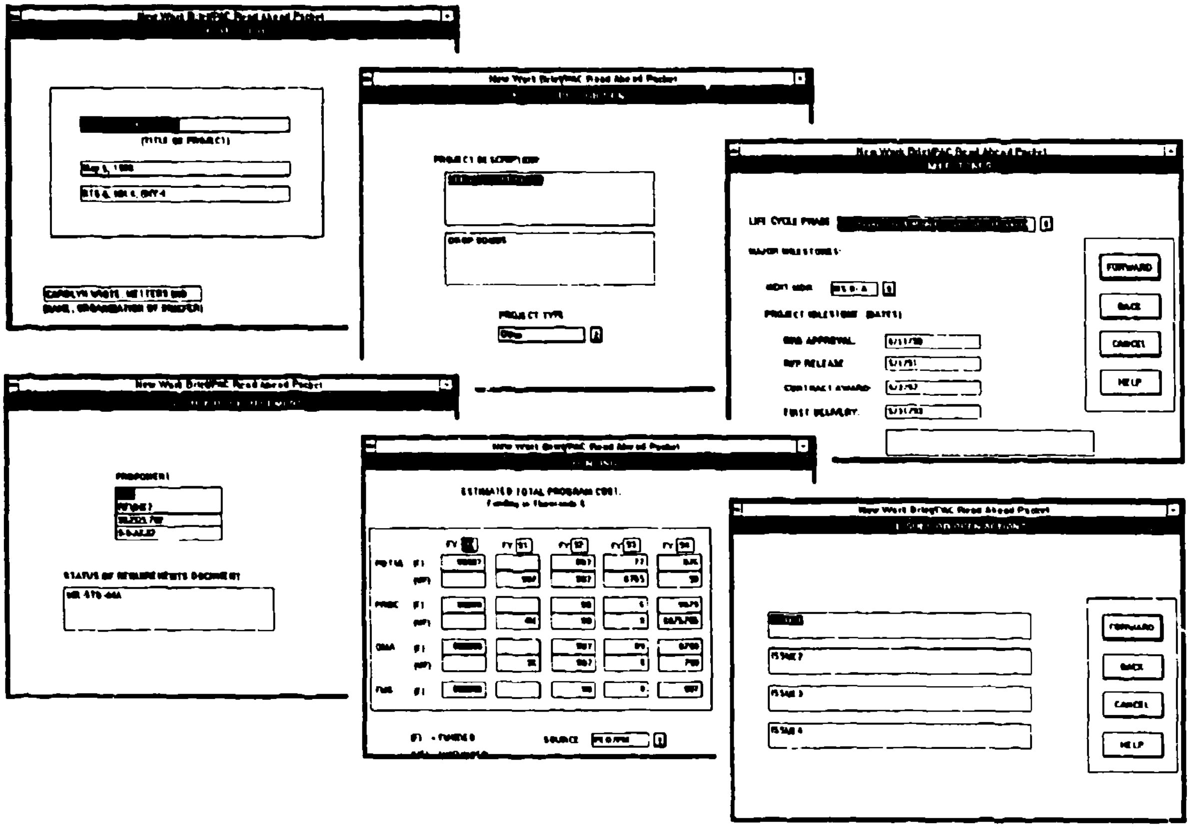


FIGURE 1. NEW WORK INPUT SCREENS

PAC Tool - A set of eight slides were developed utilizing C++, MS Project and MS Excel, to standardize the PAC briefing. The slides used for presenting a PAC briefing were titled as follows: Cover/Title Slide, Description Slide, Major Milestone Chart, Funding Summary Slide, Acquisition Summary Slide, Manpower Resource Summary Slide, Impact of New Work Slide, and Issues or Open Actions Slide. The Major Milestone Chart and its supporting Project Schedule were developed using C++ and MS Project. The Manpower Resource Summary Slide, and the Funding Summary Slide were developed using C++ and MS Excel. The remainder of the slides were created in C++ and were designed to operate in the same manner as the NWB Tool.

Project Schedule - The most challenging portion of this tool was the development of a set of generic schedule templates. The templates contain all the tasks required from the receipt of a Draft ORD to System Delivery, including Foreign Military Sales and Reprocurement. To provide maximum flexibility in schedule preparation, it was decided to provide four component areas for the templates, each with their own set of schedules. The four areas were Front End Documentation, Milestone Decision Reviews, Contracting Methodology, and Systems. The logic flow for using the templates is shown in Figure 2. A set of four dialogue boxes containing radio buttons was developed to allow the user to select a choice from each component area to build a generic schedule (see Figure 3). After all the selections are made, each component is assembled into a file in the MS Project MPX format. This format allows for a text file, in a standard comma delimited record format, to be imported to MS Project. The problem then became how to assemble these pieces, and still retain the integrity of the data and the relationships among each of the tasks. This issue was resolved by developing a subclass of the MFC CFile class that would allow a comma delimited text file to be parsed into its components, and reassembled. To maintain the linking of the tasks in each component, the predecessor field of each task had to be extracted, and the task ID number within that field had to be changed based on the task's relative location within the larger project file. The remaining information was reassembled and the task was written to disk. Once the project schedule templates are assembled into one file, the

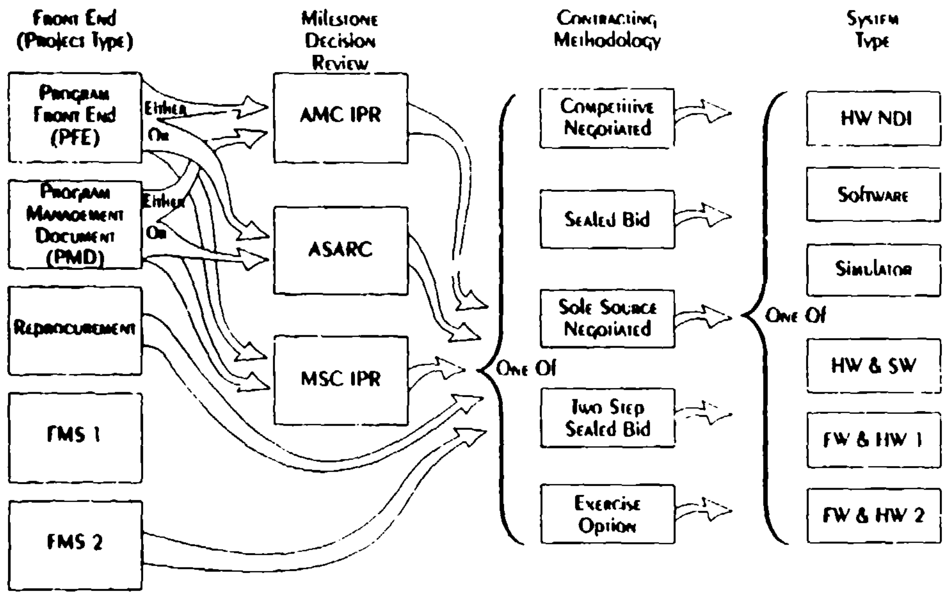


FIGURE 2. TEMPLATE LOGIC FLOW

Project team can then add, delete, or manipulate them in any way within MS Project. A macro with an associated button on the tool bar are provided so that the file is saved as both a MPP and MPX file.

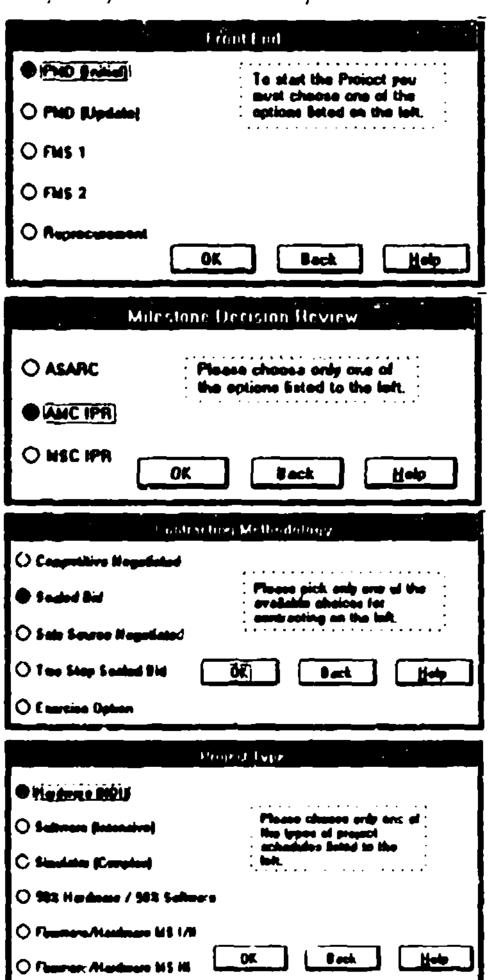


FIGURE 3. DIALOGUE BOXES

Resources and Funding -- After the project schedule is complete, the task name, and scheduled start and finish dates are extracted from the schedule. This is done using the CParseFile class developed to build the original schedule template. This information is written to a standard file format for MS Excel. An Excel add-in (XLA) and workbook (XLW) file were developed for the project team to incorporate man-hour requirements (by job discipline) and costing against the already developed schedule. To do this, information extracted from the project schedule file is imported into the task sheet where man-hours, by labor discipline, are assigned to each task. This sheet is then used to generate a quarterly man-hour report, and a man-hour cost report. Funding information for the project is entered on the Funding Summary Slide (see Figure 4). Funding information includes the type of funding and whether it is funded or unfunded. Any unused funding types are automatically deleted from the slide.

USING THE TOOLS

CCTT Program Management Tool

Tailoring MS Project - After reviewing the CCTT RFP and DOD 5000 series instructions, a few shortcomings of MS Project had to be taken into account prior to developing the CCTT schedule. First, the CCTT contract contained milestone schedules, conferences/reviews schedules, and some CDRLs with deliveries stated in terms of Months After Contract Award (MAC) or quarterly. MS Project does not recognize months or quarters as a task duration or when linking tasks together; therefore, these items had to be converted into days after contract award and number of days in a given quarter. Secondly, the Standard Calendar in MS Project, which considers week-ends as non-working days, had to be changed to a calendar that considers all days as working days. This was required because the government scheduling was

Project Funding Status ($ in 000's)

	FY94	FY95	FY96	FY97	FY98	FY99	FY00
RDTE							
6.2 Funded	1	2	3	4	5	6	7
6.2 Unfunded	1	2	3	4	5	6	7
6.3 Funded	1	2	3	4	5	6	7
6.3 Unfunded	1	2	3	4	5	6	7
6.4 Funded	1	2	3	4	5	6	7
6.4 Unfunded	1	2	3	4	5	6	7
6.5 Funded	1	2	3	4	5	6	7
6.5 Unfunded	1	2	3	4	5	6	7
OMA							
P2 Funded	1	2	3	4	5	6	7
P2 Unfunded	2	3	4	5	6	7	8
P7 Funded	3	4	5	6	7	8	9
P7 Unfunded	4	5	6	7	8	9	10
P8 Funded	5	6	7	8	9	10	11
P8 Unfunded	6	7	8	9	10	11	12
FMS Funded	3	6	9	12	15	18	21
FMS Unfunded	3	6	9	12	15	18	21
OTHER Funded	2	8	10	16	18	24	26

FIGURE 4. FUNDING STATUS SCREEN

specified in terms of 30, 60, or 90, etc., days prior to, or after an event or delivery, instead of a number of working days. Once these shortcomings were recognized and overcome, the CCTT Schedule was easy to prepare.

Schedule Preparation – The first items added to the schedule were all the major program events reviews and audits as specified in Section F of the contract. These items were linked to Contract Award with the lag time specified in the contract. CDRLs that were required prior to, or after these events were then linked to their events with their specified lead or lag times. CDRLs that were required at a given MAC or other interval, but not tied to a major event, were then added to the schedule, and linked to Contract Award. These items were then placed in the schedule at the appropriate place to maintain a time sequence of tasks from earliest to latest delivery. The linking of tasks within the schedule was accomplished using the Predecessors and Successors function within MS Project. After all items were on the schedule, the schedule was broken down into seven sub groups; Administrative, Software, Hardware, Supportability, Test, Cost, and ASARC. This concluded the preparation of the draft CCTT Schedule. This schedule contained approximately 2200 linked tasks, so that when the Contract Award date was changed, all tasks within the schedule were automatically adjusted to the correct dates.

Customizing the Schedule – After STRICOM reviewed the draft schedule, several additions were requested. The first was to provide notes for tasks to explain delivery requirements or expand on what was required for program reviews, etc. This was accomplished by using the notes function in the Detailed Task Form or Task Entry views within MS Project. Second was to add a column to assign responsibility for each task on the schedule to a project team member. To accomplish this, one of the Text fields contained in MS Project was utilized. Third was the ability to sort CDRLs out of the schedule to provide a list of CDRL deliveries in the same order as listed in the contract. This was accomplished by using a Text field and the sort capability within MS Project. Fourth was to provide the capability to look at the seven subgroups individually. This was done by creating a filter within MS Project for this purpose. Fifth was the capability to view the schedule at various levels of detail. Five levels of detail were created, using the filter capability within MS Project Level 1 displays only the Major Program Events, while Level 5 includes all tasks,

down to government review of a CDRL. Lastly was the capability to display deliveries within certain time periods, e.g., the next three or six months. Again, this was accomplished by creating a filter for this purpose.

Schedule Within a Schedule - After contract award, STRICOM requested that the contractor's proposed schedule be added to the government's already developed CCTT schedule. To accomplish this task, the contractual dates of the government's schedule were compared to the contractor's proposed dates for all tasks. If the dates for the tasks were the same on both schedules, no action was taken. When reviewing the remaining tasks, it was determined that all of the contractor's due dates for these tasks were earlier than those required by the contract. In order to depict these differences, a methodology was developed. For tasks that had different delivery dates between the government schedule and the contractor's schedule, a duplicate set of tasks was added to the schedule. The government set of tasks were "Marked" (a function within MS Project) and the term "Per Contract" was added to the task name column. These tasks' names were displayed in red print. The contractor's set of tasks was annotated with "Per IDT" in the task name column and a Flag field was added to the schedule. The Flag field, which has a "No" default, was titled "Contractor Schedule." This field was then updated with a "Yes" for those tasks which the contractor proposed to deliver earlier than required by the contract. With these additions and through the use of filters, the CCTT schedule could be viewed from the following perspectives; Government Schedule (Contractual), Contractor's Schedule (Proposed), or a Combined Government and Contractor Schedule. The later schedule allowed direct comparison to assure that none of the Contractor's proposed deliveries exceeded the delivery date required by the contract. This is the present form of the STRICOM CCTT Schedule.

NWB/PAC Tool

The NWB/PAC Tool was designed to be an easy to use tool that standardized the presentations for New Work and Project Acceptance Committee briefings. There are two separate applications within the tool, one for New Work Brief and the other for Project Acceptance Committee.

NWB - A New Work Briefing describes the requirement or nature of the new work, the source of the new work, the source and status of funding and documentation, and any significant issues. The briefing is approximately a five minute overview of a new project to determine; that the work is within STRICOM's charter, that there is sufficient information to proceed to a PAC for assessment of required resources and prioritization, the appropriate account number for man-hour and cost accounting, the STRICOM lead element responsible for the PAC briefing, the composition of the PAC Team to assist the lead element in preparing for the PAC, and the membership and proposed date for the PAC.

Creating and Editing a NWB - To create a NWB, the user selects "Create New Work Brief" from the NWB/PAC Tool's menu. The screens (Figure 1) are presented one at a time for entering the briefing information. During the creation process, the user can move between slides to make changes by utilizing the FORWARD or BACK buttons on each slide. Editing an existing New Work briefing is accomplished by selecting "Edit New Work Brief" from the Tool's menu. The screens are edited in the same manner described above for creating the briefing.

Slide Show - The capability to run a New Work Slide Show from a PC is provided by selecting "Run Slide Show" from the NWB menu. In addition, a hard copy or the slides can be printed by selecting "Print NWB" from the Tool's menu.

Acceptance - The user has found this to be an easy to use and effective tool. It has effectively standardized the presentation of all new work briefings within the organization.

PAC - The purpose of the Project Acceptance Committee briefing is to determine that the requirement is sufficiently defined and understood to warrant undertaking a significant expenditure of resources, that the draft project schedule, acquisition strategy, and estimate of required resources are realistic and executable, and what, if any, issues are associated with the project or the allocation of resources. A PAC briefing for a new project includes a description of the requirement, a milestone chart with the major project events, a summary of the funding status, a synopsis of the acquisition strategy, a detailed project schedule, a projected manpower resource summary, an estimate of the impact of the new work on the existing workload, and a summary of any issues that could impact the new work. The PAC members determine that the new work is viable and sufficiently defined and resourced with no

major issues requiring resolution. The PAC will approve/assign a priority rating to the project, approve the Project Schedule as the baseline for further work, concur with the proposed acquisition strategy, designate the lead organization for the project, and approve the formation of a Project Team and the expenditure of resources, either from within the organization or through a Support Services Contractor.

Creating and Editing the Schedule – Creation of the PAC briefing is a multi-step operation. The first step is to create the project schedule. This is done by selecting "Create PAC Schedule" from the tool's pull-down menu. A series of dialogue boxes will then be displayed (See Figure 3), each containing a series of radio buttons. The user selects one radio button, that most closely matches the project strategy, from each box. Each dialogue box has a Help button that defines the selections available and gives a listing of tasks associated with each selection. Once all choices are made, a generic project schedule is produced, which is automatically loaded into MS Project. Using the functions of Project, the user can modify the generic schedule to meet his needs. This schedule can be edited at anytime by using the "Edit PAC Schedule" menu selection.

Alternate Schedule Creation – The user can create a project schedule in MS Project directly, instead of using the dialogue boxes, and still take advantage of this tool. The schedule can be opened using the "Edit PAC Schedule" menu item. This ensures that the appropriate MS Project view file is opened with MS Project. The file can then be saved, following the appropriate file naming convention, using the save button on the toolbar. From this point, use of the tool is the same regardless of the schedule creation method used. This provides the flexibility to begin using the tool at any point of a project's life cycle.

Creating and Editing Travel Information – While working with the project schedule in MS Project, travel cost information can be entered. This is the second step of the PAC briefing creation process. This feature allows the user to enter travel costs against any task. Three buttons have been added to the MS Project Tool Bar for this function. The first button changes the Project View so that a travel cost column is displayed next to the task name, duration, and scheduled start and finish dates. This allows entering travel cost for any task desired. The second button will display all tasks that contain a travel cost entry. The third button is for printing a travel report. The report contains all tasks, with their start and finish dates, that have a travel cost entry, plus the total cost of travel for the project. Editing the information in the travel column is done through MS Project.

Creating and Editing Resource and Funding Information – Before creating the manpower resources for a project schedule, the project schedule must be fully developed and saved in MS Project. After this is done, selecting "Create Funding and Resources" from the Tool's menu will open an MS Excel application for entering the following information: Man-hours by job discipline against each project task, and funding by funding type for the project. This is the third and fourth step in the creation of a PAC briefing. The input screen for entering funding information is shown in Figure 4 and the input screen for manpower information is shown in Figure 5. From this application two reports can be generated, after the required information has been entered. These are: a Quarterly Man-Hour Utilization Report, and a Man-Hour Cost Report by Discipline. (Note: To accomplish the man-hour loading and costing for man-hours, two tables have been pre-loaded into the tool that are transparent to the user. One is the job disciplines for STRICOM, a generic Support Services Contractor, and some outside agencies, e.g., CECOM and TECOM. The other is the average cost per hour for a government employee and for a support contractor.) To edit the PAC resources or funding data, select "Edit PAC Funding and Resources" from the Tool's menu.

Creating and Editing Other PAC Slides – The final step in creating the PAC briefing is to generate the remaining slides required for the PAC brief. To create these slides select "Create PAC Slides" from the Tool's menu. The titles for these additional slides are: Cover Slide, Acquisition Strategy, and Issues or Open Actions. Entering information on these slides is accomplished in the same manner as for NWB. To edit these slides select "Edit PAC Slides" from the Tool's menu.

Slide Show – The capability to run a PAC Slide Show from a PC, is provided by selecting "Run PAC Slide Show" from the Tool's menu.

Data Transfer – After the PAC brief has been presented and the project approved, the Project Schedule and Project Resources become the baseline for the program. The capability to upload the Project Schedule and Project Resources to the STRICOM MIS has been provided

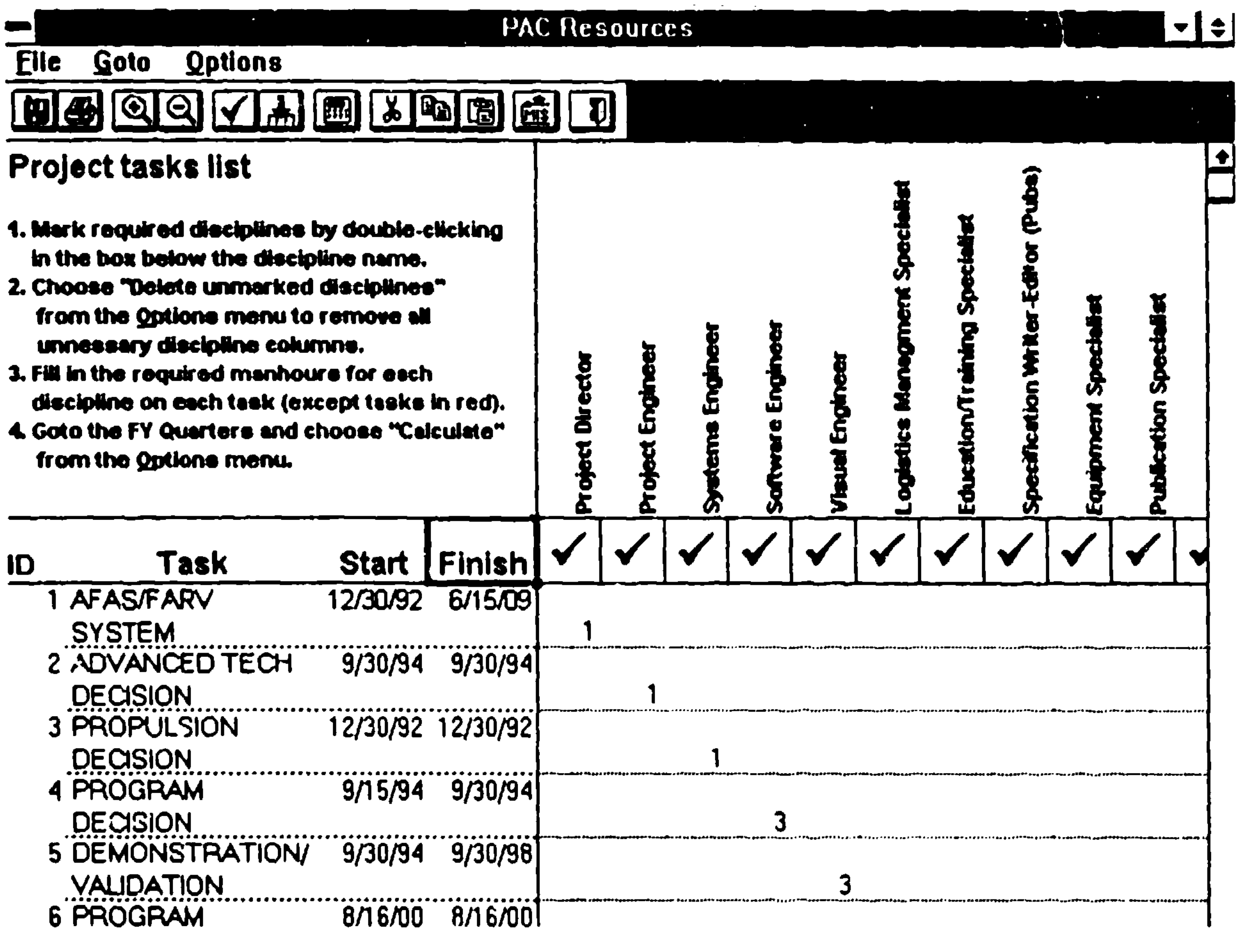


FIGURE 5. MAN-HOUR INPUT SCREEN

through the use of the "Transfer to MIS" selection from the Tool's menu.

Acceptance - The user has found this tool more difficult to use than the NWB. This has been true in two areas; creating the project schedule, and assigning resources, man-hours by job discipline, to each task on the schedule. Although the assignment of resources is a time consuming effort the benefits derived by management and the user's customers have been apparent. This will be discussed in the next section. The PAC Tool has met its goal of standardizing the briefing presentations of all new projects within STRICOM.

BENEFITS DERIVED FROM TOOL USAGE

CCTT Program Management Tool

While using this tool, numerous benefits were realized. One benefit was the ability to discern discrepancies within the RFP prior to Contract Award. When Section F of the contract, the SOW and the CDRLs were evaluated and put on the schedule, scheduling discrepancies were readily apparent. These discrepancies involved CDRL due dates versus Section F events, duration of events in the SOW versus Section F events, and CDRL review dates that went past the Program Review they were supporting. This provided the government the ability to correct the RFP prior to contract award, which greatly reduced the number of contract modifications or change proposals that would have been necessary if the discrepancies had not been discovered prior to contract award. Another benefit was the ability to add option packages to the schedule, that were linked to the date of exercising the option. By changing the start date of the option, "what if" drills could be accomplished to determine the best date to exercise the option. Another benefit, similar to the exercise of option benefit, was the ability to move major events; e.g., SRR, PPOT, IOTE, and determine the impact on the overall schedule. This could be

accomplished because all events that were applicable to the major event were linked to it, so that, when its date was changed, all of the linked events would change also. The ability to predict "peaks and valleys" in terms of resources required throughout the contract was another benefit realized. A continuing benefit is the ability to track all deliveries and determine, at any time, the status of the contract. This includes the ability to project early, on-time, or late completion based upon the current status of deliveries. The addition of the Change Comment Sheets to the tool has proved invaluable in determining when and why dates were changed and by whom.

NWB/PAC Tool

NWB Tool - The main benefit derived from the NWB was the optimization of the process used for accepting new work into the STRICOM organization.

PAC Tool - Like the CCTT Project Management Tool, the PAC Tool has provided numerous benefits to its users. The overview of a new project, in terms of funding status, man-hour requirements and cost, schedule, and acquisition strategy, has been an immediate benefit derived from using this tool. Besides providing a detailed schedule, this tool has provided the ability to plan the required man-hours by job discipline to accomplish new work coming into the organization. The ability to develop a fact-based cost estimate for other Project Managers and Program Executive Offices for the procurement of training devices has also been provided by this tool. The cost estimate is based upon the number of man-hours derived by the tool, times the cost of each man-hour. Because of the tool's ability to display man-hours required, it can be used to determine whether sufficient in-house personnel exist to accomplish a given function, or whether those functions should be handed-off to a Support Services Contractor. Although the tool has only been in use for three months, it has the capability to be used to justify requirements for additional personnel spaces, or funding levels for a Support Services Contract. These are the major benefits that have been realized in the short time the PAC Tool has been utilized.

APPLICABILITY TO OTHER USERS

CCTT Program Management Tool

The scheduling capability of this tool combined with its filters and sorting, and the addition of Action Item Tracking, and Change Comment Tracking make this a very powerful Project Management Tool. Whether this tool is MS Project or some other COTS software is unimportant, the capability to use software to plan and execute a program is the important fact. This tool, or any other with its capabilities, is applicable to defense contractors, other major subordinate commands, other services, and any organization that procures or manufactures goods and services.

NWB/PAC Tool

This tool has been specifically designed to meet the needs of STRICOM. In its present form it is applicable to any defense procurement agency. Minor modifications would have to be made to the schedule templates, job discipline table, and cost table in order to meet the needs of other procurement agencies, and provide the same capabilities to them as the tool presently provides to STRICOM. With further modifications to the job discipline and cost tables, and additional development in MS Excel to provide the ability to cost materials per task, and apply Overhead, G&A, and Material Handling costs to labor and materials as appropriate, this would be an excellent tool for proposal costing.

SUMMARY

Both the CCTT Program Management Tool and the New Work Brief/Project Acceptance Committee Tool were developed with COTS software as their base. They have both proven to be easy to use and have provided tremendous benefits to STRICOM in the short time they have been in place. These tools have made the job of Program Management a little easier and more effective. The Tools have provided a factual basis for estimating required man-hours and cost for a project. Whether these specific tools are used, or others like them, matters not. The capability they provide is what matters.

A HYPERTEXT TOOL TO SUPPORT DEVELOPMENT OF FLIGHT SIMULATOR SPECIFICATIONS

Edward A. Martin, Ph. D.
Training Systems Branch
Integrated Engineering and Technical Management Directorate
Wright-Patterson Air Force Base OH 45433-7800

and

John F. Lethert
Air Mobility Training Systems Division
Training Systems Program Office
Wright-Patterson Air Force Base OH 45433-7800

ABSTRACT

A long-standing problem in the acquisition of flight simulators has been the clear communication of requirements through the specification process. There are numerous reasons for this, including obfuscation by technical jargon, fragmentation of requirements within a specification, and a human inclination to adopt "cut and paste" approaches which may reflect the requirements of a precedent system more than those of the current system. This paper discusses an Air Force initiative to develop a hypertext-based generic guidance specification for flight simulators that attempts to address these problems. Each generic specification paragraph includes hyperlinked recommendations and rationale for specification language, verification, and options. Knowledge -- based upon systems engineering principles -- is embedded in logic that guides the author through the development of specifications. Since this guidance specification is embodied in a software tool that makes it relatively easy to use, the expectation is that it will be used. If it is used, the documents produced will reflect the high degree of standardization imposed by this guidance specification. It will provide a clear alternative to less-disciplined cut-and-paste approaches, and emphasize sound systems engineering practice. Standardized format and vocabulary will help avoid misplaced information and inconsistent interpretations. Localization and integration of requirements will minimize conflicts.

ABOUT THE AUTHORS

Edward A. Martin is a Technical Specialist in the Training Systems Branch of the Integrated Engineering and Technical Management Directorate. He is responsible for the development and deployment of technical processes for flight simulator acquisition and training system integration, and the infusion of new technologies into simulator and training products by coordinating the technical efforts in the branch and advising Aeronautical Systems Center (ASC) program offices. He earned the BS in Electrical Engineering from the Virginia Military Institute, the MS in Electrical Engineering from Syracuse University, and a PhD in Biomedical Engineering from the Ohio State University. He is a member of the IEEE and AIAA, and is a registered professional engineer. Dr. Martin has over twenty-five years' experience in aerospace simulation, both in industry and with the Air Force.

John F. Lethert is the Chief Engineer for Air Mobility Programs in the Training System Program Office. In this role, he provides technical advice and leadership for the C-17 Aircrew Training System (ATS), the C-17 Maintenance Training Devices, the C-141 ATS, the KC 135 ATS, the Special Operations Forces ATS, the Simulator for Electronic Combat Training, and the T-3 Aircraft. He is the Air Force Representative on the Distributed Interactive Simulation Steering Committee. Mr. Lethert holds a Bachelor of Electrical Engineering from the University of Minnesota and the MS in Electrical Engineering from the Ohio State University. He is a member of the IEEE, AIAA, and Association of Old Crows, and is a registered professional engineer. He has over twenty years of experience with simulators and training systems.

A HYPERTEXT TOOL TO SUPPORT DEVELOPMENT OF FLIGHT SIMULATOR SPECIFICATIONS

Edward A. Martin, Ph. D.
Training Systems Branch
Integrated Engineering and Technical Management Directorate
Wright-Patterson Air Force Base OH 45433-7800

and

John F. Lethert
Air Mobility Training Systems Division
Training Systems Program Office
Wright-Patterson Air Force Base OH 45433-7800

INTRODUCTION

Purpose of Guide Specifications

Guide specifications (formerly Military Primary or "Mil-Prime" Specifications) were initiated as part of an Air Force Systems Command effort to streamline the acquisition process and provide greater latitude for contractor innovation (Gordon, 1985; Industry News, 1989). Guide Specifications are expressly designed to force tailoring to the specific application through the use of blanks inserted in the requirements, and are the vehicles of choice recommended by MIL-HDBK-248B to support the tailoring mandated by the acquisition streamlining policies and procedures of DODI-5000.2. Specification tailoring is not necessarily easy, and can actually impose a greater burden on those who write and issue solicitations to thoroughly understand the operational or training environment and develop realistic requirements based upon the intended application (Gershanoff, 1988). While they do provide suggested specification wording and guidance, Guide Specifications cannot serve as cookbooks for devices as varied and complex as flight simulators. The new Flight Simulator Guide Specification, which is to replace AFGS-87241A, incorporates many unique features and concepts to ease the tailoring process while supporting (and encouraging) a structured systems engineering approach to facilitate the clear definition of requirements.

The Flight Simulator Guide Specification

The new Flight Simulator Guide Specification is developed to support both government and industry tailoring of specification documents. It is envisioned that the government will primarily use this to develop the Systems Requirements Document (SRD), and that contractors will use it for generating the Type A and Type B specifications. For this reason, the Flight Simulator Guide Specification *format* is designed to produce hard-copy documents with section and paragraph arrangements that agree with the requirements of MIL-STD-490B for these types of specifications. This is done to facilitate a consistent evolution from SRD to System Specification to Complex Item Requirements Specification (CIRS), and to provide for continuity in the decomposition and location of requirements across the family of specifications for a given flight simulator.

The *content* of the Flight Simulator Guide Specification has also undergone a major change to reflect the latest Air Force simulation requirements. Guidance is now included for aircrew trainer types ranging from part-task trainers to weapon system trainers to mission rehearsal devices. Both fixed-wing and rotary-wing aircraft simulators are covered. There is an increasing demand for simulators that are affordable in large quantities, and which can be placed in small facilities that are not necessarily environmentally controlled. This market requires scaled-down simulators possessing something less than full fidelity simulators; therefore explicit guidance has been added for dealing with

"selective-fidelity", i.e., the tailoring of fidelity to the specific application. Visual/sensor database generation system accuracy and throughput specification guidance is included in the Guide Specification to support mission rehearsal simulations. Process guidance is added to suggest processes that should be established to better define and document requirements; this guidance is intended to flag key specification interfaces with development processes.

As with predecessor Guide Specifications, recommended language is provided for function, performance, verification methods, and options. These recommendations use terminology consistent with MIL-STD-490B, and provide wording that defines each requirement in terms amenable to verification. Consistent use of both requirement and verification language is emphasized and encouraged by conventional approaches, as well as the hypertext authoring tools discussed later.

KEY PRINCIPLES

An effective Flight Simulator Guide Specification should serve as a tool that facilitates the process of translating user training requirements into specific characteristics of a simulator device. This process starts with very general requirements such as: "practice air-to-air combat, air-to-ground weapon delivery, and emergency procedures at their regular training bases and operating sites." As these top-level requirements are evolved into flight simulator devices, questions that are asked repeatedly are -- "What do you really mean?" "Is this what you really intend?" "Will this system meet your needs?" These questions are asked repeatedly during the requirements' development process, source selections, development of the device, and test of the device -- and these questions are repeated when it is necessary to modify a device. The specifications should capture the answers to these questions as the acquisition progresses. Four key principles can make this happen. These are:

a. Standardized language and terminology.

b. Object oriented decomposition.

c. Use of a structural model.

d. A logical top-down specification development process.

Standardized Language and Terminology

A famous children's book uses the phrase, "I meant what I said, I said what I meant" (Suess, 1940). This principle has generally been the exception rather than the rule in specification development; nevertheless, it is fundamental to writing good specifications. Consider the following examples drawn from the current AFGS-87241A -- "The DRLMS[1] shall be designed to satisfy the accuracy specified in the following paragraphs..." (3.9.1.1), "The image generator shall have the capability to retrieve and process..." (3.7.2.4) and, "The engine systems to be simulated shall include but not be limited to..." (3.3.7.2). Does the first sentence mean that the DRLMS will be designed to meet the requirements, but not fabricated to meet them? Does it mean that no testing will be required? Does the second sentence mean that the image generator need retrieve and process data perhaps one time out of 100 requests? Clearly these were not the authors' intentions. In the third sentence the engine systems are listed and its author intended "include, but not be limited to" as a contingency statement in case something was overlooked -- but what the sentence really says is that even if a completely accurate list is provided, the contractor must provide something else. These comments may appear as "nit-picking", but the authors personally know of several occasions where long government and contractor discussions have revolved around such minor discrepancies in terminology. Another major language problem encountered is purely semantic -- words have different meanings to different people. For example, does the word "platform" refer to an aircraft carrying a radar system or to a motion platform? What do the terms "user friendly", "high fidelity", and "adequate to train" really mean? The new Flight Simulator Guide Specification addresses these terminology and semantic problems with five basic rules.

The first rule is, "The system shall do it." In the first example above, what was obviously intended was, "The DRLMS shall meet the accuracy requirements of the following subparagraphs." In the second example the intention was, "As the simulated aircraft moves across the gaming area, the image generator shall retrieve and process..." So why not say it? In almost every instance words such as "shall be

[1]Digital Radar Landmass Simulator

designed to" or "shall be capable of" are used to provide variety in writing style. However, this also leads to variety in interpretation. Specifications should communicate requirements -- they do not have to be interesting reading.

The second rule is, "Say it once." MIL-STD-490A permitted a large degree of interpretation regarding the organization of various requirements. Because of this, there has been a tendency to cover the same item in different sections of the specification. Instructor controls are one classic example. In the past, instructor controls were often discussed both in an instructor console section and in sections such as the simulated aircraft and simulated environment. When a requirement is stated in more than one place, authors tend to write the requirement differently in each place. In addition, a specification document will often be changed during the development process, and it is extremely likely that a requirement may be changed in one place and not another. If a requirement is stated in one place -- and one place only -- the likelihood of internal conflicts arising in the specification is reduced. The *architecture* of the Guide Specification, which is discussed later, is one mechanism used to achieve the goal of stating a requirement in only one place. In addition, the recommended language isolates discussion of simulator controls from the effects of those controls. As an example, refer to Table 1 and Table 2. The language recommended in Table 2 defines which flight controls are to be included in the simulator (and the appearance and tactual fidelity of those controls), but does not attempt to state how the simulated aircraft should respond to those controls; the aircraft response is handled in one place -- under "Air Vehicle Dynamics".

The third rule is, "One name -- one meaning." This avoids the "platform" problem discussed earlier. To facilitate this rule, the Guide Specification includes definitions that are easily accessed using some of the Guide Specification Toolkit features discussed later.

In addition, the Guide Specification encourages the clear definition of key performance characteristics (such as "freeze", "record", and "replay") right from the outset of a specification's development. This is necessary to ensure consistent understanding and implementation of these characteristics across all simulator subsystems. In previous specifications, performance characteristics have often been called out as "one-liners", addressed differently in different sections, or sometimes not mentioned at all. Where performance characteristics have not been well defined, time has been lost in lengthy deliberations to arrive at a consensus regarding the "real" user requirement. These situations have also provided additional opportunities to arrive at inappropriate implementations and ultimate user dissatisfaction. To help avoid these problems, the Guide Specification includes the following:

a. Simulator modes are defined up front. These are defined to be, "a simulation state or collection of simulation states which represent fundamental ways of operating the simulator from the viewpoint of a crewmember or operator." Examples include normal operating mode, networked operation, maintenance mode, and part task operating modes.

b. Up-front definitions are provided for events such as freeze, crash, halt, and malfunctions. The malfunction definition includes the requirement that, "Other systems shall respond to this change of system state in a natural manner in accordance with the design criteria... i.e., the malfunction effects shall propagate through the simulated system in a manner analogous to failure propagation through the real system." This also forces software models into closer correspondence with the real world; the benefit of this is discussed in the next section.

c. Guidance is provided so that activities such as record, replay, stabilization, crash override, set and reset of various simulator conditions, and mission time management are all explicitly defined from the start.

Table 1

Definition of fidelity levels for crewstation instrumentation, controls, and displays.

Replicated	Shall be in the correct location, and have the appearance and feel of the aircraft equipment as defined by the approved design criteria. Shall be interfaced with the simulator computational system such that the computer can identify the state of the controls, and drive the instrumentation and displays in accordance with this specification.
Depicted	Need not be in the exact location, nor have the appearance or feel of the aircraft equipment. Shall be interfaced with the simulator computational system such that the computer can identify the state of the controls, and drive the instrumentation and displays in accordance with this specification.
Inert	Shall be in the correct location, and have the appearance and feel of the aircraft equipment as defined by the approved design criteria. Need not be interfaced with the simulator computational system.
Pictorial	May be a static picture of the aircraft equipment, but shall be of the size and in the location defined by the approved design criteria.
Not Required	No representation of the aircraft equipment is required.

Table 2

Examples of recommended language, verification methods, guidance, and embedded examples extracted from the "Flight Controls" simulation requirements.

Requirement.	The __1__ flight controls shall be __2__. Gearing shall be in accordance with the approved design criteria. Simulated force feedback at the flight controls shall be __3__. __4__ shall be __5__.
Verification.	This requirement shall be verified by inspection and test. Inspection shall verify compliance with appearance and location requirements of the flight controls and controls located upon these flight controls. The tolerances specified in this paragraph shall also apply to relevant dynamic tests conducted in accordance with...
Requirements Guidance.	Blank (1) should identify the flight controls to be included, and should be tailored to the application. This would typically be "wheel, column, and pedal" for a classical transport aircraft... In lower fidelity applications, all flight controls (e.g., pedal) may not be required. Blank (2) should state the required level of fidelity for the flight controls, which would typically be "replicated, and have displacement in accordance with the approved design criteria." For a very low fidelity device (where a simple joystick might suffice)... Blank (3) should define the force simulation required, in accordance with the following guidance: Full-Fidelity Simulation Of Traditional Mechanical Flight Control Systems: In this case, high-quality force feedback is required. Put the following into blank (3)... Lesser Fidelity Simulators: This might apply for devices intended for uses such as procedural refresher training of mission-qualified pilots... If full-fidelity force-feedback is not required, put the following into blank (3)... Blanks (4) and (5) are intended to capture the fidelity requirements of the switches, buttons, and any other controls located on the flight controls ...
Verification Guidance.	The "Verification of Flight Controls" paragraph is written for a full-fidelity simulation of a traditional mechanical flight control system, and should be tailored downward where requirements call for lesser control loading fidelity. Tests or demonstrations should be retained for any quantitative requirements such as gearing, control envelope, and control free response. ...
Process Guidance.	If a level of fidelity "representative of the aircraft force feedback" is specified in blank (3), a process should be defined which results in: (1) the force-feel simulation being prototyped and demonstrated with the user, and (2) the documentation of the acceptable force-feel transfer characteristic, and its incorporation into the approved design criteria.
Example.	Where a lesser-fidelity device is being acquired for supporting practice of highly procedural tasks, substantial cost savings might be realized by reducing the fidelity required of the force-feel system. For example: "Flight Controls. The stick flight control shall be replicated, and have displacement in accordance with the approved design criteria. The pedal flight control shall be inert and need not pivot, but shall be adjustable fore and aft (using the Rudder Pedal Adjustment on the Control Pedestal) over the full range specified in the approved design criteria. Gearing shall be in accordance with the approved design criteria. Simulated force feedback at the flight controls shall be representative of the aircraft force feedback, but need not provide full tactual fidelity and may be realized using passive control loading devices. Force feedback shall be in accordance with the approved design criteria. The two-position gun trigger on the stick shall be inert. The following stick controls shall be replicated: a. Four-way aircraft trim switch. b. Weapon release button. c. Fore/aft/down auto-acquisition switch."

Simulation requirements are often very difficult to describe. Quantitative requirements are used in the Guide Specification where possible, but there are many instances where it is extremely difficult -- or simply not feasible -- to write quantitative requirements. Qualitative terms, such as those dealing with required levels of fidelity, are defined at the outset so that all parties will have a common understanding of their meaning. For example, the level-of-fidelity definitions provided in Table 1 appear in the Guide Specification prior to their use in lower-level paragraphs. This leads to the fourth rule, "If fidelity must be described qualitatively, use a series of well-defined adjectives and define it consistently." Table 2 provides some examples regarding the use of the level-of-fidelity definitions in recommended specification statements.

The final rule is, "Distinguish between process and product." Process is very important -- as evidenced by the emphasis on process throughout the acquisition community. A sound process is essential to produce the device in a manner that converges the user expectations and the final product. Previous specifications have often referred to processes in the form of requirements to conduct reliability and maintainability or integrity programs, requirements to identify malfunctions at design reviews, and the specification of tests to be performed at the contractor's plant versus on-site. The problems with this approach are:

a. There is often overlap with the Statement of Work. This leads to the risk of violating the "say it once" rule.

b. There is no closed loop; the process may

An agency is preparing a System Specification with minimal security information available. The Statement of Work tasks the contractor to perform a System Security Engineering Program. The specification language recommended by the Guide Specification in this case would be:

3.3.9 System Security. The vulnerabilities of the simulator shall be minimized.

The Process Guidance in the Rationale section of the Guide Specification would recommend that the Statement of Work require the contractor to update the specification when new information becomes available as a result of the System Security Engineering Program.

Process Guidance: A System Security Engineering Program should identify threats and vulnerability. It must also require appropriate contractor interface with the accrediting or certifying authority. It must provide appropriate information for this specification when necessary.

After the security issues are resolved, the specification is updated to read:

3.3.9 System Security. The following vulnerabilities of the simulator shall be minimized with the countermeasures indicated:

a. Software viruses. A commercially available virus protection program shall be incorporated in the simulator such that all software files are checked on installation.

b. Access to classified data by personnel not cleared to the appropriate level. Use of the system shall require a login password and unique id with authentication data. The probability of a guess shall be .000001 or less. There shall be no group id's. A Discretionary Access Control (DAC) system shall insure that classified data is accessed on a need- to- know basis. In addition The computational system shall be declassified by one: 1) removable hard disc drives and 2) an overwrite algorithm. The overwrite software shall overwrite the computer memory locations a minimum of three times

c. Physical damage or destruction. An alarm shall indicate access to the cockpit unless authorized at the instructor console.

d. Access to the KY 601 Panel when the simulator is unattended. The KY 601 Panel shall be easily removable when the simulator is not in use.

The highest level of information processed in the simulator shall be SECRET.

Figure 1. Separation of process and product in the Guide Specification. An example drawn from the System Security section.

change the characteristics of the device, but the changes are not captured in the specification.

c. Authors may inadvertently discuss process when they mean product; this produces ambiguous meanings.

d. Process must be tailored to the particular device, the particular program, and the particular acquisition agency.

The Flight Simulator Guide Specification deletes all process language from the specification text. It then includes process guidance in the rationale section. This rationale section is intended to support development of the Statement of Work. Fig. 1 provides one example wherein a process is recommended to update the specification once the necessary information becomes available. Table 2 provides another example wherein a process is recommended to quantify level-of-fidelity requirements and then include these in the approved design criteria.

Object Oriented Decomposition

Standardized language and terminology are essential, but the architecture of the specification itself can be just as important in communicating clear requirements. The traditional flight simulator specification architecture is a product of the evolution of the flight simulator. Flight simulators began as devices to provide instrument training. They were essentially a simulation of the dynamics of the air vehicle. As technology improved, simulators became much more complex. The industry added motion systems, visual systems, complex avionics and electronic combat simulations, Digital Radar Landmass simulations (DRLMS), and complex instructional systems. As systems were added, simulator manufacturers' engineering departments added experts to deal with each of these functional areas. Government personnel became likewise specialized. These experts tended to focus on their specific functional areas rather than the whole system; this led to specifications being organized around functional areas. As systems became more complex, this contributed to ambiguity in requirements.

The traditional specification breakout illustrated in Fig. 2 arose from this focus on functional areas. How are the requirements described under this traditional approach? The air vehicle section covers aircraft performance. The avionics section covers the avionics computer that interfaces with the electrical system, which is in turn covered under the air vehicle. Aircraft radar is partially covered under avionics and partially under Digital Radar Landmass (DRLMS). The visual and radar sections both deal with terrain and the same portion of the earth's surface; these requirements differ only in regard to appearance attributes at different portions of the electromagnetic spectrum. Features, such as an enemy radar, must appear both on the visual system and the aircraft radar -- and are covered in both sections. The electronic combat section not only covers enemy radar illuminating the aircraft, but also aircraft displays covered under avionics. Visual, electronic combat, DRLMS, and the air vehicle all cover weather effects. Instructional requirements are frequently covered in multiple places, and the instructor console requirements interface with everything else. This traditional breakout leads to a great deal of overlapping requirements, and affords ample opportunities to violate the "say it once" rule. In addition, with this traditional breakout there is no direct correspondence to the real world. This makes it more difficult to relate simulator requirements directly with user requirements, which are often expressed in real world terms.

The Flight Simulator Guide Specification takes a different approach. It incorporates an architecture that corresponds more to the real world. All requirements on the primary air vehicle are in a simulated air vehicle section. This section parallels ASC's Air Vehicle Guide Specification. The remainder of the real world is represented by the simulated environment. This includes other vehicles (entities), the terrain, magnetic variation, celestial objects, and electromagnetic radiation. Interactions are natural and implicit; vehicles hide in terrain and weather attenuates radar signals. Translating the military mission into simulator requirements is simplified and straightforward, relative to the traditional breakout. The Flight Simulator Guide Specification approach is illustrated in Fig. 3.

We might view a simulation as a series of models representing the aircraft and a series of databases and models representing the rest of the world. In nature there is natural interaction between the vehicle and the environment. For example, if a window is put into the vehicle the pilot can see the world. However, merely putting a window in the simulator gives the pilot a nice

view of the closest wall. Systems are needed to produce the interaction between the simulated air vehicle and the environment. The Guide Specification calls these systems cue generators. This section of the Guide Specification includes such requirements as motion systems, image generators, and visual displays.

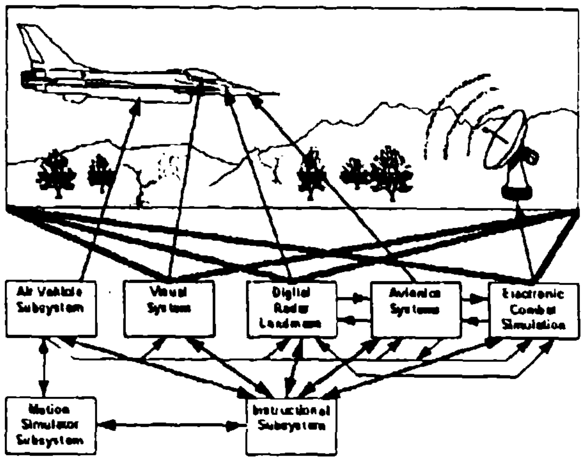


Figure 2. The traditional Flight Simulator Specification architecture.

Traditional specifications typically include the requirement for instructor consoles or systems in one section, but disperse many of the interface and interaction requirements to the relevant subsystems. The Guide Specification avoids this scattering of simulator control requirements in the following way. In the simulated air vehicle and environment sections, the term "commanded" is used to denote that a system is to respond to an instructor, operator, or automated control input. However, details regarding interactions and implementation are all included in the simulator control section. This section also deals with the display of information an instructor may require to operate the system or evaluate trainee performance.

The final section of the Guide Specification (which does not appear in Fig. 3) deals with system support. This includes requirements such as those for a Training System Support Center or a database generation system.

Use of a Structural Model

The Software Engineering Institute or SEI (1992, p. 5) offers the following definition for a structural model. "A structural model is a pattern for specifying and implementing software system functionality. It reflects system engineering decisions about partitioning and coordination."

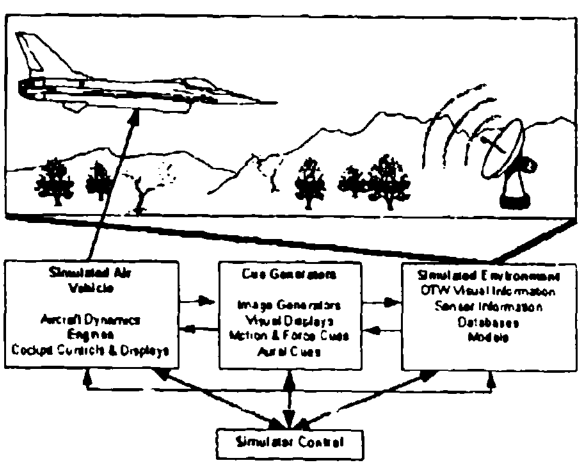


Figure 3. The architecture employed in the new Flight Simulator Guide Specification.

The structural model concept was used on B-2, C-17, and Special Operations Forces Aircrew Training System simulators. The partitioning strategy suggested for a structural model is an object-oriented decomposition wherein software objects are related to real world entities (Software Engineering Institute, 1992). This inspired the object-oriented partitioning strategy discussed in the previous section. The object-oriented architecture of the Guide Specification should, in itself, facilitate the specification and implementation of software system functionality. However the "Computer Resources" section of the Guide Specification goes further, and explicitly requires a structural model as follows:

> "All simulation software shall be object oriented. The software architecture shall have three levels; they are: a) executive level, b) subsystem level, and c) component level. Each subsystem level shall model the systems on the real aircraft and each component shall model the components of the aircraft being simulated (e.g., fuel pumps, turbines, etc.). The executive level shall coordinate the subsystem level. The subsystem shall manage a group of components at the component level so that they will behave as a unit. The component level shall be concerned only with computation. Each subsystem shall have no direct knowledge of other subsystems. All information shall be transferred from the memory

locations where each subsystem will place its data. The components shall have no knowledge of the outside world except through input or output parameters. Any systems outside the aircraft such as radars, missiles or other aircraft shall also be simulated using structural modeling and object oriented design."

Additionally, use of a structural model will help enforce consistency across all simulator subsystems. One of the basic concepts of the structural model is the use of common software templates. This should impose consistent software implementation by designers.

The Training Systems Program Office (ASC/YT) is developing a Structural Model Handbook with the SEI. This is intended to communicate concepts and design approaches to the flight simulation community as a means to help transition this technology. Once complete, a reference to the handbook will be included in the guidance section.

A Logical Top-down Specification Development Process

The Flight Simulator Guide Specification is a lengthy, complex product. It was started as a guide for writing:

a. A System Specification for a simulator procured as a stand alone device.

b. A Complex Item Requirements Specification (CIRS) for a simulator procured as part of an aircrew training system -- or a standalone system where there was a requirement to expand and clarify the System Specification.

The Guide Specification is intended for government or contractor use. Current policy at ASC requires that contractors write specifications and that the government write an abbreviated requirements document called a System Requirements Document (SRD). Since an SRD will be expanded into a Systems Specification and Systems Specifications may be expanded into CIRS, the same overall organization should be used with each lower level of specification containing more detail. Thus the Guide Specification can be used as a guide for SRD preparation. It is essential, however, that the specification development process proceed in a logical manner.

Specification or SRD development should not begin until after an Instructional Systems Development process has identified the key requirements of the device. Then, the first step in developing the specification or SRD is to name the simulator and identify its purpose. This is accomplished by filling in the blanks in the Guide Specification paragraphs shown in Fig. 4. Thereafter, as also shown in Fig. 4, key physical constraints are identified. The constraints should match the specific application; for example, if the device is to be portable and fit into an office, there should be constraints imposed on size and weight.

The next step is to identify the major vehicle and environmental characteristics that must be simulated. In addition, there must be a check for consistency between the device's purpose and its physical attributes. Is the right portion of the cockpit required? Are all needed environmental characteristics provided? Are physical constraints consistent with other requirements?

The process proceeds through succeeding steps in a similar manner. As it progresses, several facts become evident:

a. The order in which paragraphs should be written becomes less obvious.

b. Multiple authors can work on paragraphs independently.

c. Certain paragraphs may be inappropriate to the level of specification being written.

The Guide Specification Toolkit, discussed in the next section, includes a "User Guide" utility to facilitate this top-down specification development process.

1. SCOPE. This specification establishes the requirements and associated verification methods for ___.

Requirements Guidance: *Fill in the blank with device name, nomenclature (if available), and aircraft represented (if applicable).*

3.1 System Definition. The __1__ is intended for use __2__ in __3__ at __4__. It shall comply with all requirements of this specification.

Requirements Guidance: *Fill in blanks as follows:*

1. *Fill in device name*
2. *Fill in who uses and their initial qualifications*
3. *Fill in task(s) to be accomplished*
4. *State where the device is to be used*

3.2.2 Physical Characteristics. The simulator shall consist of: ___1___. The simulator ___2___.

Requirements Guidance:

1. *This blank should describe the principal physical entities (e.g. cockpits, crew stations, instructor stations, motion bases, etc.) which make up the simulators well as the physical relationships between them.*
2. *This blank should describe any limitations on the dimensions or weight of the principal physical entities. The sentence may be deleted if there are no dimensional or weight limitations.*

Care must be taken to avoid unnecessary size or weight restrictions. Dimensions must be consistent with other requirements. (e.g. the height must be consistent with visual display requirements).

Figure 4. The first steps to the top-down development of a specification.

THE GUIDE SPECIFICATION TOOLKIT A HYPERTEXT-BASED SYSTEM

The new Flight Simulator Guide Specification includes an authoring system toolkit that runs under Microsoft Windows. This provides a familiar graphical user interface for specification development that features the same sorts of menus, dialog boxes, and "point-and-click" controls found in other Microsoft Windows applications. To promote usability and acceptance, every effort is made to make the user interface in this toolkit as consistent as possible with other Windows-based programs so that the interface controls respond in the way expected by the user

This Guide Specification includes the Windows-based toolkit for several reasons. First, the specification includes a large volume of specially formatted text that must be tailored significantly. Editing with a mouse-based, graphically-interfaced, authoring system can be considerably easier than working with a simple text editor. Second, Microsoft Windows is a widely-used, well-accepted, graphical mouse-based interface. As such, the interface should be familiar to most of the targeted users of this toolkit. Third, it is easy to import text from many of the other Windows word processors into this application. Finally, the Windows graphical user interface inherently supports the implementation of hypertext links.

Hypertext is useful because it can greatly expedite navigation through large volumes of text, and facilitate the access of data from a variety of directions (Midford, 1989). The essence of hypertext is computer support for links within and between documents (Conklin, 1987). Say, for example, you were reading an encyclopedia article entitled "Primates", and at the bottom of the article the text read "see Mammals." If you wanted to know more about mammals, you would flip to that article, read through it and find the pertinent information. In a hypertext environment, the computer would automatically look up the article on mammals for you when you clicked on the word "mammals"

(which would be designated as a "hotword" in the text). Such hypertext techniques are widely used in Windows-based programs.

The software toolkit provides a range of productivity enhancement tools that would otherwise be unavailable in a strictly hardcopy format. Some are discussed below.

Support for Standardized Language and Terminology

Software-controlled linking is exploited in order to foster common usage and understanding of terms. Two tools are provided: (1) a glossary of definitions and acronyms, and (2) graphics to convey key concepts in conjunction with the definitions. The hyperlinked glossary, illustrated in Fig. 5, provides quick and convenient access to definitions of terms from anywhere within the toolkit. The capability will also be provided to automatically write these definitions and acronyms to Section 6 of the hardcopy specification documents in MIL-STD-490B format. Supplemental graphics are used to illustrate and clarify key underlying concepts. These are accessed by clicking an icon that appears on those toolkit pages having such graphical explanations. Fig. 6 and Fig. 7 provide an example by way of graphics used to illustrate the relationships among modes, activities, and events. An icon appearing on any of the toolkit pages dealing with "Performance Characteristics" associated with these topics will instantly link the user to a graphic such as shown in Fig. 6. Clicking on the appropriate control will then bring up further detail, as illustrated in Fig. 7. If desired, definitions for any of the events shown can be obtained by clicking on that event's name. These tools provide not only convenient and rapid access to definitions, but serve to explain concepts in ways that cannot be accomplished nearly as conveniently in hardcopy format. Use of these tools should facilitate consistency in the specification language and the interpretation of that language.

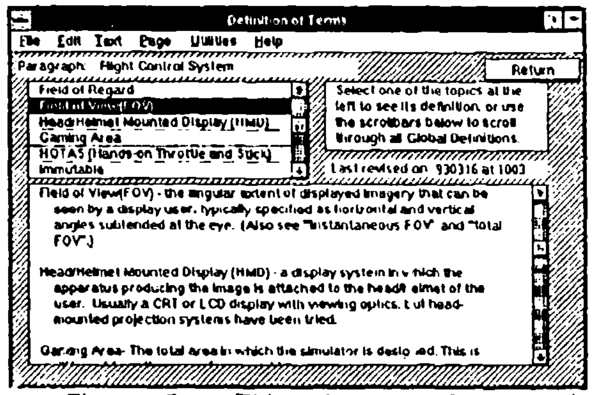


Figure 5. This glossary of terms is conveniently accessed through a pull-down menu from any where within the Guide Specification Toolkit.

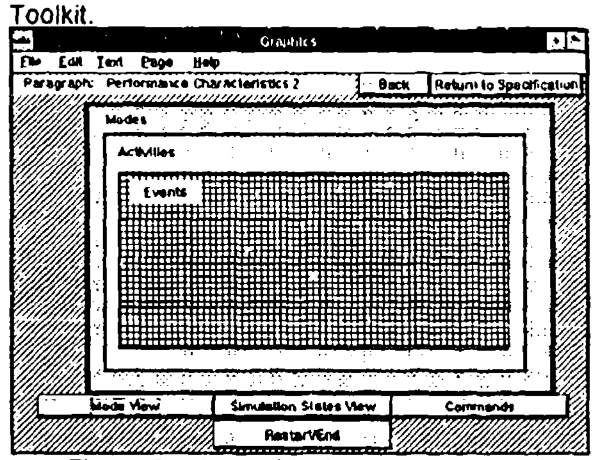


Figure 6. A graphical representation of the relationships among simulator modes, activities, and events that is displayed in response to selecting the "Mode View" button. This page is accessed by clicking an icon on the Guide Specification pages dealing with these topics.

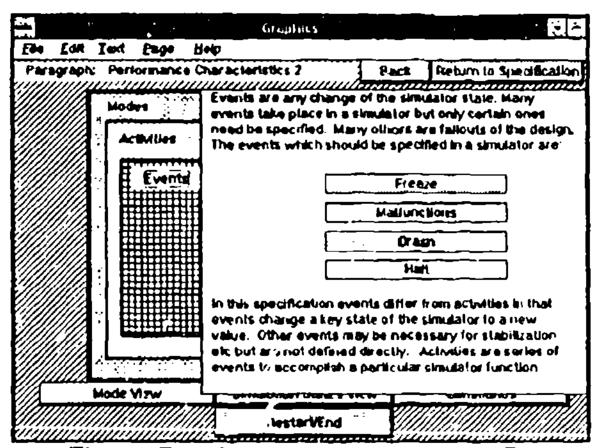


Figure 7. Clicking on the word "Events" causes this definition to be displayed. Clicking on any event name will display the definition of that event.

Efficient Access to Material

The table of contents is designed as a multilevel menu to hasten the process of locating a desired paragraph. A hierarchy is employed so that no paragraph titles beyond three levels of indenture appear in the main table of contents (Fig. 8). This provides the user a top-level view of the organization, and facilitates identification of the desired topic area. Clicking on appropriate "hotwords" (designated by the boxes surrounding the text) permits the user to either go to a lower level of detail in the table of contents, or directly to the desired paragraph. For example, clicking on "3.7.3" in Fig. 8 will cause the main subparagraphs under "Simulated Air Vehicle" (Fig. 9) to be displayed. In order to proceed to the subparagraphs under "Air Vehicle Crew Systems", the user would then click on "3.7.3.6" in Fig. 9 -- which would bring up the screen shown in Fig. 10. At this deepest level-of-detail, only paragraph titles appear as hotwords. Clicking on a paragraph title anywhere within the Table of Contents will immediately access the designated page. Clicking on "Flight Controls" in Fig. 10 would take the user to the Flight Controls specification "page" as illustrated in Fig. 11.

Each guide specification "page" incorporates all the guidance paragraphs for a specific topic such as "Flight Controls". The paragraph numbers (both System Specification and CIRS, in accordance with MIL-STD-490B) and name appear at the top of the page. The relevant text is displayed in a scroll box in the center (this text can be edited or copied by the user). Underneath

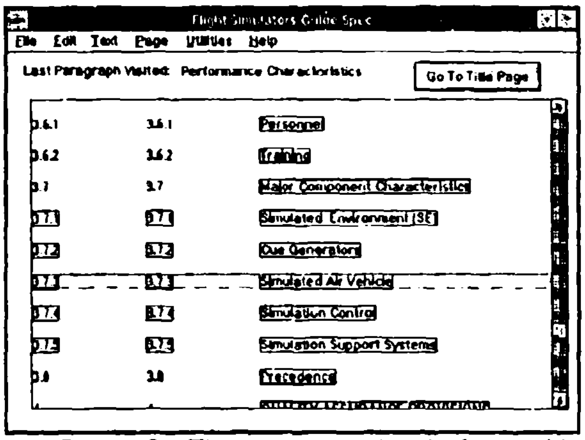


Figure 8. The uppermost level of the table of contents pages. Clicking on a paragraph name links directly to that paragraph. Clicking on a hot paragraph number links to the next level-of-detail table of contents menu.

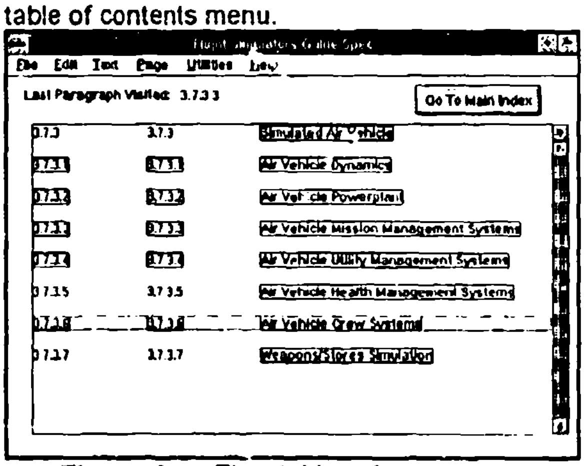


Figure 9. The table of contents menu reached by clicking on "3.7.3" at the preceding level.

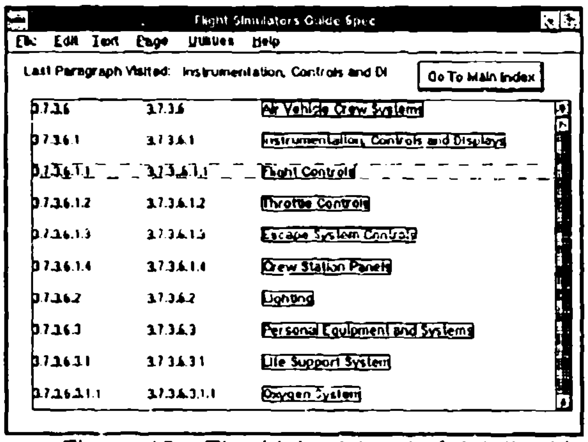


Figure 10. The highest level-of-detail table of contents menu. This is reached by clicking on "3.7.3.6" at the preceding level.

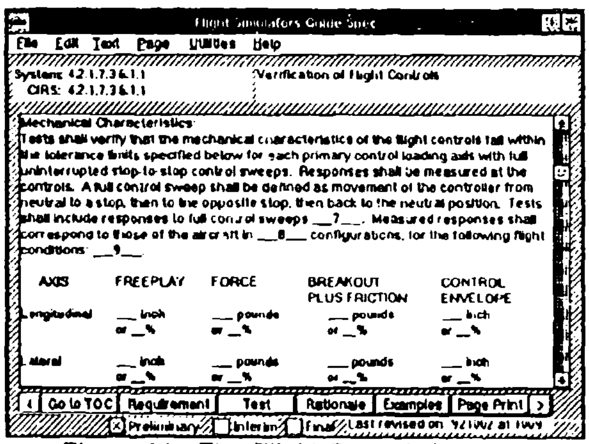


Figure 11. The Flight Controls "page" of the Guide Specification Toolkit. The page as shown is opened to the verification paragraph.

the text box is a row of buttons by which the user can display:

- the recommended requirement or verification language (buttons "Requirement" and "Test", respectively).
- rationale and guidance ("Rationale" button).
- examples of completed paragraphs.

Other buttons permit the user to obtain a hardcopy output of the contents of the entire page ("Page Print") or to navigate within the toolkit (the "<" and ">" buttons move to adjacent pages, while the "Go To TOC" returns the user to the Table of Contents). Under the buttons are three checkboxes that are used by the specification author to assign the maturity level for each page. In addition, a time stamp is provided which indicates the last time the page was revised.

Support for a Logical Top-down Specification Development Process

"User Guide", illustrated in Fig. 12, is a user-selectable utility that is provided to facilitate the top-down specification development process discussed earlier. Essentially, User Guide utilizes a hierarchical structure, and guides the user to develop the specification in accordance with that structure. The User Guide's display page provides summary information for all relevant paragraphs regarding their place in the hierarchy (or "level"), their maturity status (null, P or preliminary, I or interim, and F or final), and their last revision date. Hyperlinks are provided which allow the user to navigate directly between specification pages and the utility display page, so that the specification paragraphs can be readily accessed and edited in the proper order.

Each Guide Specification paragraph may be assigned a "level", which consists of a single letter and a three digit number. The three digit number determines the order in which specification paragraphs should be written within a "group" -- i.e., lower numbers should be written first. The letter is the "group" designation. Group "A" must be developed first. Groups "B" through "Z" can then be developed in parallel. The maturity of a paragraph within a group cannot be greater than all paragraphs with lower numbered levels. The User Guide utility reads the maturity status from the checkboxes at the bottom of the relevant pages (see Fig. 11), and determines whether this rule was violated. If so, the user is provided a warning along with contextual guidance regarding how to proceed.

The User Guide is an optional tool for specification writers. It identifies a preferred order in which specifications should be written and provides indications when the order is violated. The specification authors are allowed to omit paragraphs and to edit paragraph order before using it. Its use then forces requirements' decisions to proceed logically -- on the basis of previous, more fundamental decisions.

PLANS AND PROGRESS

The Guide Specification Toolkit has already undergone a number of in-process reviews, and is now at a point where it can be more widely distributed for evaluation. Revision of the specification content is nearing completion, and is currently undergoing internal reviews. It is anticipated that the Guide Specification will be released for industry review during the summer of 1994. We will welcome any and all feedback from those reviewing our product -- and are especially interested in feedback regarding:

a. Content.

- ⇨ Did we follow our rules? For example, were requirements stated in only one place and did we say what we meant?
- ⇨ Do the recommended requirements language and guidance make sense?
- ⇨ Do the recommended requirements and

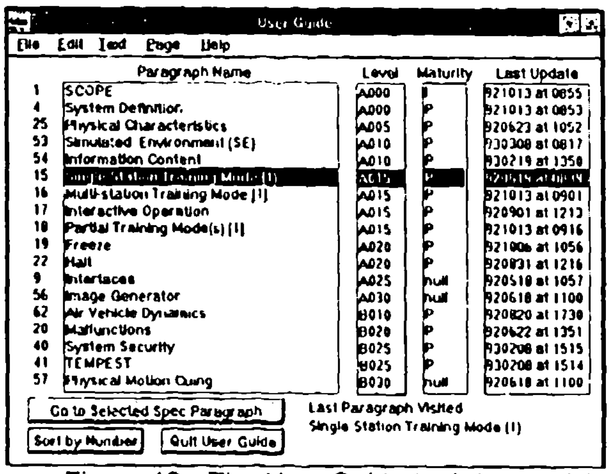


Figure 12. The User Guide tool that assists in developing specifications in a logical, top-down manner. Paragraph names are shown sorted by level. Clicking the "Sort by Number" button would order paragraph names in the order that they appear in the Guide Specification.

verification language and guidance yield requirements that can be evaluated? Is there a definitive basis for acceptance or rejection? Does it make sense?

b. Architecture.

- ⇨ Does the way in which the requirements were allocated against the architecture make sense?

c. Guide Specification Toolkit.

- ⇨ Is the Windows-based toolkit easy to use? Does it add value to the specification-authoring process?
- ⇨ Are there toolkit features that should be implemented in a different way?
- ⇨ Are there additional toolkit features needed, or enhancements that would improve the process?

SUMMARY

Our desire is to build a useful tool that will help both us and the simulation industry. We hope that the emendation of content and architecture truly leads to improved specifications, and that the added Guide Specification Toolkit features serve to facilitate the tailoring process. It is our expectation that making this Guide Specification easy to use will enhance its application, and lead to greater standardization in specification format and language. We need your views, and look for serious feedback from the industry review.

ACKNOWLEDGMENTS

The authors would like to thank Lt Brett Borghetti for his contributions supporting the Guide Specification Toolkit discussion.

REFERENCES

AFGS-87241A, U.S. Air Force Guide Specification for Flight Simulators. 9 Jan 1989.

Conklin, J. *Hypertext: An introduction and survey.* Computer, 21(1), pp. 17-41, Sep 1987.

DOD Instruction 5000.2. Defense Acquisition Management Policies & Procedures. Part 6, Section Q. *DOD standardization program,* and Part 10, Section C. *Acquisition streamlining,* 23 Feb 1991.

Gershanoff, H. *Liar's dice.* Journal of Electronic Defense, p. 14, July 1988.

Gordon, J.K. *USAF writes new specifications to ease its prime contractor role.* Aviation Week & Space Technology, pp. 83-84, 5 Aug 1985.

Industry News. *AVIP demands better electronic systems.* Defense Electronics, p. 34, Feb 1989.

Midford, R. *Vendors probe hypertext for repair crews.* Federal Computer Week, p. 26, 20 Mar 1989.

MIL-HDBK-248B, Acquisition Streamlining. Section 8. *Streamlining tools and techniques,* 9 Feb 1989.

MIL-STD-490B, Preparation of Program-unique Specifications. ~~7 December 1992~~ (draft).

MIL-STD-680B, Standardization Program Requirements For Defense Acquisitions. 1 Mar 1990.

Software Engineering Institute. *An introduction to structural models.* Presented to Interservice/Industry Training Systems and Education Conference (I/ITESC), Training Systems Program Office (ASC/YT), Wright-Patterson AFB OH 45433, ASC 92-2719, 1 Nov 1992,

Suess, Dr. *Horton hatches the egg.* Random House, New York, 1940

LIFE CYCLE COST MANAGEMENT FOR COMPREHENSIVE TRAINING SYSTEMS

Leland O. Singer
Boeing Defense & Space Group
Huntsville, Alabama

Ralph Smith
Lockheed Aeronautical Systems Company
Marietta, Georgia

ABSTRACT

In several current weapon system programs, Training System development is integral to the prime system project. High level training needs are established at a Training System level while specific requirements are developed under the prime contract using Instructional System Development (ISD) or a similar process. This affords substantial flexibility in developing a Training System configuration that yields true life cycle cost effectiveness. However, this can only be exploited if a practical means of determining the cost is used during specific requirements development. The interesting management challenge is that the range of system design options is almost limitless. The complexity associated with doing cost analyses, and the potential for disjointed and incomplete analysis is high. Clearly, some method of creating an organized, thorough and efficient cost analyses is needed.

This paper portrays a life cycle cost based process that evaluates all aspects of training cost. The process provides an organized method of analyzing and recording all development, production, and support resources required for the Training System and their associated costs.

The process uses data Input Tables that include such things as: the skill mix required to operate and maintain the prime system; the prime system deployment schedule related to the need dates for trained personnel; and types of media, support resources and instructors required for each course. These and similar inputs are related to each other through Process Tables that include Cost Estimating Relationships.

This methodology provides traceability of costs to training needs and support resources and a structured process that prevents aimless, unreliable analyses.

ABOUT THE AUTHORS

Leland O. Singer received a BS in Physical Sciences from Kansas State University and an MA in Logistics Management and Supervision from Central Michigan University. He served in the Air Force as an instructor pilot, research pilot, aircraft maintenance officer and program manager. He joined Boeing in 1979 and has served in programs such as the B-52 and B-1B aircraft, the Roland and Avenger Air Defense systems and Training Systems for the B-1B, E- 6A and Roland. He is currently developing a Life Cycle Cost management capability for the F-22 Training System. Mr. Singer is recognized as a Certified Professional Logistician by the Society of Logistics Engineers and has a broad academic background with over 20 years of experience in product development, production and support.

Ralph Smith is Manager, Design To Cost/ Life Cycle Cost for the F-22 Advanced Tactical Fighter program at Lockheed Aeronautical Systems Company. He is responsible for integrating all aspects of DTC/LCC management of the air vehicle, Support System and Training System and coordinating the Boeing, Lockheed Marietta and Lockheed Ft. Worth Company efforts in this area with the U. S. Air Force System Program Office. Mr. Smith holds a Masters in Business Administration from Pennsylvania State University. He has over 12 years experience in the application of cost models and Life Cycle Cost analysis techniques as engineering tools to aid designers in optimizing the cost-effectiveness of their systems.

LIFE CYCLE COST MANAGEMENT FOR COMPREHENSIVE TRAINING SYSTEMS

Leland O. Singer
Boeing Defense & Space Group
Huntsville,Alabama

Ralph Smith
Lockheed Aeronautical Systems Company
Marietta,Georgia

INTRODUCTION

In the current defense business environment, substantial emphasis is being applied to minimize all aspects of program cost. This is no less true for training costs associated with defense programs. One can argue that the majority of operations and support activities associated with many weapon systems are related to training.

In this paper a Training System is defined as the collection of training assets needed to support employment of the prime system. We view the training process as an organized whole that carries comprehensive cost implications for development, production and usage of the training assets.

In many past programs, training capabilities were developed over many years, by several different contracting agencies, and in many separate contracts. However, current weapon system development programs require concurrent development of the Training System as a part of the prime system contract. This includes the requirement to propose and contract for the Training System's development as a part of the prime contract. Many of these programs require that Training System development, production and operations and support be estimated as part of the original proposal effort. This can be a particularly daunting task because specific training requirements are usually created as a part of the contracted development effort. This means that Training System developers must estimate the cost of products for which no specific requirements exist.

One typical way of estimating cost under conditions of uncertain specific requirements is to use historical, parametric data from previous, similar efforts. However, training managers and analysts have often experienced significant difficulty in finding relevant and complete data regarding what the total cost of Training Systems has been on previous programs. We have found a vacuum in our professional knowledge base regarding what comprehensive Training Systems have cost in the past. This leaves us with a considerable problem in determining what they should logically cost for future programs.

This paper deals with thought processes and methods that can be used to understand the total cost implications of developing, producing and supporting training resources and then using those resources to conduct training throughout a weapon system's life cycle.

LCC AS THE CONTROL PARAMETER IN TRAINING SYSTEM DESIGN

Cost Ceilings As Economic Realities

The traditional emphasis on cost control has evolved into the more severe concept of *cost ceilings* in many, recent weapon system procurements. Cost has been established as a make or break factor in program survival. While other requirements do exist and have relevance when assessing program viability, cost considerations have taken on a greater relevance than they may have in the past. Examples can be found where capability has tended to be downgraded to meet cost objectives. Space Station Freedom is one current case in point.

In short, if programs are not viewed as affordable in today's business climate, they tend to die quickly. This is true for Training Systems as well as for prime systems. If funds are limited, program managers understandably tend to believe that the prime hardware is more essential than its Training System. If sacrificing or limiting Training System funds is deemed necessary to save a weapon system, it will normally happen.

Since cost ceilings in Training Systems are now economic realities, then training analysts, designers and managers must adopt the philosophy that cost control is the single, most important design parameter in meeting training requirements.

Certainly training requirements in and of themselves are significant. However, if the cost of meeting those requirements is not affordable, they will not get met. It is then, important to establish requirements that we can afford to meet.

Clearly, if a Training System development effort is to be successful, we must have viable tools to reliably estimate the cost of providing the needed training capability. These tools must also give visibility into what causes the estimated costs to be what they are. They must also show us how to influence the product design to meet the cost requirements.

SYSTEM DESIGN FLEXIBILITY IS BOTH GOOD AND BAD

There is an encouraging, positive sign that goes along with the emphasis on cost control in many current Training System acquisitions. The final system requirements and system designs can be optimized to meet both the requirements of the instructional analysis and the cost objectives. While broad training concepts are generally outlined in a system level specification, the details about how to meet these concepts are left to the development team. This means that we can enjoy substantial flexibility in establishing a system concept that meets both training and cost objectives. Many trade-offs can be made in assigning training objectives to the most cost efficient media and instructional venue. During the instructional analysis, Computer Based Training (CBT), training devices, conventional classroom presentations and hands-on training on the prime hardware can all be considered. Additionally, assignment of training requirements to different training locations can provide further flexibility in developing the most efficient system configuration.

Flexibility in determining the system configuration has a down side also. Selecting the optimum configuration from an endless number of options can be very confusing. Both the costs associated with a system configuration, and its ability to satisfy overall training objectives must be determined if we are to get the best solution. For example, it might appear straight forward to determine the cost and training effectiveness of developing and administering a single 6 hour CBT course. But, how do we know that this single course fits efficiently into the overall curriculum needed for the complete weapon system?

TRADITIONAL LCC METHODOLOGY AND WHERE THE HOLES ARE FOR COMPREHENSIVE TRAINING SYSTEM DEVELOPMENT

LCC analysis techniques are not new in defense system acquisition. Over the years many tools and many specialized processes have been developed to deal with this subject. However, these processes have not always dealt accurately or efficiently with the unique problems associated with complete Training Systems. Several factors have combined to create the need for improved LCC methods.

1. Many different training requirements exist to support a complete weapon system. They include operator and maintainer training (at all maintenance levels) as well as training for personnel that support the direct employment of the weapon system (e.g., supply and fire protection personnel). Training must address initial qualification training, skill upgrade and continuation training throughout an individual's career. These are specific training elements which contain unique performance requirements that must eventually be identified to a very finite level of detail.

Traditional LCC approaches tend to view systems at a high level which may obscure effects caused by changing seemly minor details. For instance, if one hour of conventional classroom courseware is added to the curriculum, the development cost could increase by perhaps $10000 with negligible production cost. However, the cost of conducting that hour annually for 500 students a year for 20 years could easily exceed $2,000,000. Traditional LCC models may not reveal the impact of this change within in a

total curriculum of several hundred hours of instruction. The processes in traditional models may not be sensitive to subtle differences between cost drivers during the development, production, and operations and support phases of the training program. This paper proposes methods that can deal specifically with these effects.

2. The total scope of training is large and spread over many years. The associated costs are high enough that it is difficult to envision a top level parametric that relates the cost of the Training System to the weapon system in an understandable way. The historic data is fragmented and may not relate closely to the conditions under which the new Training System must operate. Traditional LCC methods tend to rely heavily on historical data for validation. Without comparable data for complete Training Systems being readily available, it is difficult to calibrate many existing models to correlate well to complete system developments.

3. Many factors that influence training costs are not well established early in the weapon system development process. Such things as operational mode summaries and the Logistics Support Analysis will not be accomplished until several years after the weapon system enters full scale development or, at best, are very immature in the early stages of the program. Factors such as the number of training locations, the number and complexity of maintenance tasks, and the types of training assets required are often unknown at the time cost estimates and contracts are established. The best we can hope for is to have an idea of the final requirements and their associated LCC. Traditional LCC approaches rely extensively on broad parametrics to create estimates at a high level. If we cannot develop the broad factors that LCC models demand, we have difficulty developing reliable results.

4. The quantity and variety of the factors that drive the LCC of a Training System establish a level of a complexity that make it difficult to create organized, repeatable cost analysis using traditional approaches. If the factors that determine the costs are not apparent to the reviewer (or the analyst, for that matter) our confidence in the numbers will suffer. Rigorous analysis of the technical and performance characteristics of a proposed system can aid in improving the quality of an LCC estimate, but many traditional methods are applied at too high a level to encourage such an analysis.

This paper presents approaches to dealing with these characteristics found in traditional LCC analysis and presents an alternative to correct the short falls.

LCC METHODOLOGY AS AN INTEGRAL PART OF THE TRAINING SYSTEM DESIGN

Paradigm For Successful LCC Analysis In Training Systems Design

The approach presented here discretely identifies each element of the Training System and determines the characteristics of the element that influences cost. We then apply Cost Estimating Relationships (CER) to translate the cost driving factors into monetary terms. This sounds very conventional. However, our approach dramatically increases the direct involvement of analysts by requiring them to diliberatly establish each relevant factor and CER for each situation rather than adapt a standardized, automated model to the situation. This approach requires the analyst to clearly identify all factors that the analysis addresses and document each element in a rigorous pattern. This documentation takes the form of simple textual or mathematical expressions and ensures that the analytical process is clearly disclosed and repeatable. Automation is not an essential feature of our process. Any automation that is used, is largely to simplify computation and presentation of results through mechanization of manual methods.

We describe the process as *static, discrete or deterministic* modeling to distinguish it from the probabilistic, predictive processes that are often associated with LCC techniques. Figure 1 shows a high level view of this concept. Most of the CERs that we use are simple, linear connections. These simple relationships tend to be easier to understand than probabilistic relationships that infuse the notion

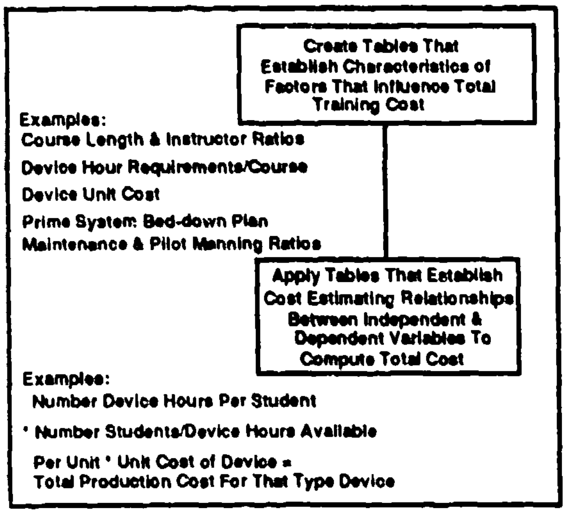


Figure 1 Training System LCC Model Architecture

of uncertainty into the results. The simple relationships tend to be easier to visualize and more intuitive to understand than probabilistic functions. They also tend to be easier to "validate" in the absence of comparable historical data because they are simple and intuitive. One can explore the model piece by piece and validate the reasonableness of each piece without getting lost in the often complex dynamics of more traditional approaches.

Figure 2 shows components found in typical Training System LCC analyses. The estimating methodology that is used to determine the cost values is key to both the accuracy of the results and the efficiency of conducting the analyses.

The method presented in this paper is to first postulate a complete system configuration for the Training System. At least all the elements shown in Figure 3 should be described with as much specificity as practical after considering the maturity of the prime system design and its deployment and employment concept. A combination of sketches, block diagrams, course listings and other descriptive data should be created that will allow estimators to envision the configuration that is under consideration.

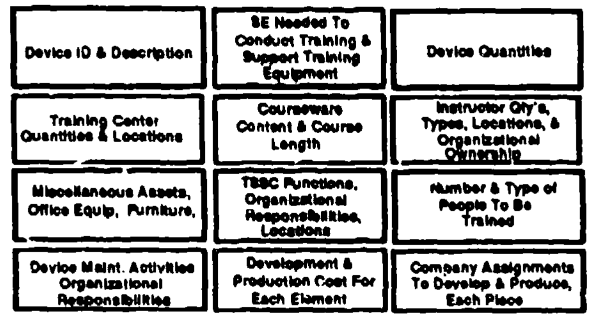


Figure 3. LCC Related to the Entire Training System Description

Next, estimators further describe detailed elements of the complete Training System. Figure 4 is an example of input data that identifies resources that are required to conduct a course. Figure 5 provides another example of this type of data where we identify the number and skill levels of maintenance personnel that are needed to support the system. Figure 6 demonstrates a bed-down schedule for the prime system that is used with Figure 5 datato determine the number of trained personnel and when they are needed. Figure 7 shows a way to identify training device costs and Figure 8 shows instructor support resource requirements. The rest of the elements that are needed to estimate the total LCC are established individually in a similar fashion.

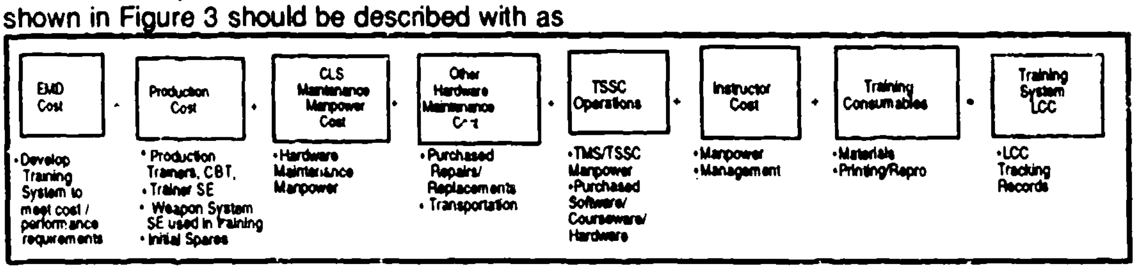


Figure 2 Typical Training System LCC Elements

Course Title	Cource Number	Flow Time (Days)	Contact Hr's	Course Repeat %	Course Remediation Hr's/Student	Course Wash Out %
Structural Repair	100-1001	30	150	10	10	5

Major Course Support Resources		Course Characteristics			
Instructor Type	Hr's/Course. Required.	Max. Class Size	Frequency For Each Student.	Responsible Agency	Prime Course. Location
Composite Engineer	110	6	1 Time	Contractor	Tech Training Center
Welding Tech	40	Training Aids		Training Consumables	
Training Devices	Hr's./Crse./ Stud. Req'd	Composite Panels	SE (Used as Prime Media)	Item	$ / Stud. / Course.
Struc. Dyn. Sim.	12	Repair Station	Heat Blanket	Adhesive	1000
Computer Based Devices		Aluminum Stock	TIG Welding Machine	Fabric	500
Student Station	20	Grinder		4041 Rod	100

Definitions

Training Device - Training equipment using significant amounts of actual or simulated A/C or support mission equipments

Computer Based Device - General purpose automated training equipment using few or no actual or simulated mission equipments

Training Aids - Small, low cost items that are not operational or operate manually (e.g. video players, projectors, cut away LRU's, white boards, wall charts, component mock-ups, etc.)

Training Consumables - Materials provided by the course administrator that are consumed in the conduct of the course (e.g. adhesives, sheet metal, rivets, programmed texts

SE (Used as Prime Media) - Support Equipment used in the conduct of Operations & Maintenance Courses e.g. M32A-60 used to conduct courses). SE used to maintain training devices is not included here

Course Remediation Hours Per Student - Average hours per student that remedial instruction is required for student to successfully complete course on first attempt

Course Repeat Percentage - Average percentage of students that must repeat the entire course to successfully graduate

Course Washout Percentage - Average percentage of students who never successfully complete course

Course Length

Flow Time - Total time in days, the course is designed to run

Contact Hours - Total active instructor monitored hours designed into course

Frequency for Each Student - Number of times a year student is required to repeat same course

Primary Course Location - Type of training center normally conducting training

Responsible Agency - Organization directly responsible to budget & conduct course

Figure 4 Course Support Resource Table

At this point, finite, obvious elements of the Training System have been identified and simple, discrete cost parameters have been established. The analyst has been required to discretely address each element and make informed judgments about the cost of each element. These activities require that the LCC analysts be very knowledgeable of Training System activities and discrete cost estimating, but they do not require detailed knowledge of techniques like linear programming, cueing theory, improvement curves, statistical analysis and others that are often used for sophisticated cost estimating. While these techniques all have demonstrable worth, some organzations may not have experienced practioners available who can use them effectively.

Skill Name	Skill Code	No. Pers. Per A/C	Ovhd. %	Min Alwd. Per Site
Dir Maintenance				
Avionics	452X0	0.2	0.1	2
APG	452X4 N	0.4	0.2	4
Elect/ECS	452X5	0.1	0.1	2
Armament	462X0	0.2	0.1	4
Propulsion	454X0 A	0.05	0.1	2
Direct Pilots:				
Pilot	1115A	1.4	.1	3

Figure 5 Required Manning Table

Year	Loc.	1	2	3	4	5
Cum. A/C		4	8	16	24	34
A/C Per Yr.		4	4	8	8	10
Test		2	1	2	2	
FTU		2	1	2	2	2
Base 1			2	4	2	2
Base 2					2	6

Definitions
Table Used to Record Planned A/C Per Year Location

Figure 6 Aircraft Bed-down Table

Next, the analysts must establish the CERs that exist between the various input data. Figure 9 is a simplified example of a CER for the cost of entry level training for a fuel system mechanic

Development $	Unit Production Cost $
500,000	100,000
Annual Operation & Support $	
10,000	Material
20,000	Labor
5,000	Purchased Services
1,600	Replenishment/Spares
500	Shipping

Figure 7 Training Device Cost Input Data

Many types of CERs will be needed to model the LCC of the complete Training System. They will be unique to the specific Training System configuration and how the input data is structured. There is no single, optimum solution to how the analyst chooses to create the input data. For example, if the number of people required in a given skill code is known, that number could be entered directly in the model. The number could also be computed on the basis of so many people per prime vehicle. The CERs that are needed in each model are obviously different.

As with the case of the input data structure, the analyst is still establishing discrete elements of cost and has not yet attempted to produce the final answer. This is a simplifying technique to prevent the analyst from being overwhelmed by the apparent magnitude and complexity of the complete analysis.

OVERALL MODEL ARCHITECTURE

The analyst establishes the system configuration, the element cost estimates, and the CERs using discrete estimates for each piece. When all the element estimates and CERs are in place for a given configuration, it is time to compute the total LCC and document the results. Only at this point will the analyst truly understand the complete implications of a given Training System configuration. This may be a source of some concern if the pressure for intermediate results or short term answers is high. Until the end of the process, the final result will not be known or cannot even be speculated on with any degree of certainty. When all the individual pieces are in place, significant complexity may exist and a large amount of number crunching is normally required to add all the elements together to obtain LCC. If the Training System configuration is large and complex, some type of automated calculation capability may be in order to save time and expense and minimize computational errors. Tools like spread sheets, data base managers and hard coded custom programs can be used for all steps in the LCC analysis. These include entering, storing and retrieving data, computing total costs, and preparing the results as tabular or graphic presentations.

Instructor Type	Resp Agcy	Inst. Num	Man Yr Cost ($9K)	TDY Exp ($90K)	O'Seas Exp ($90K)
APG	1	1	50	10	20
Hydrau	0	2	53	8	15

Definitions
Resp. Agcy. - Org. responsible to Provide Instructor
Inst. Num. - ID Number
Man YR Cost - Annual Ops & Supt Expense
TDY and O'Seas Exp - Annual Ops & Supt Expense

Figure 8 Instructor Support Resource

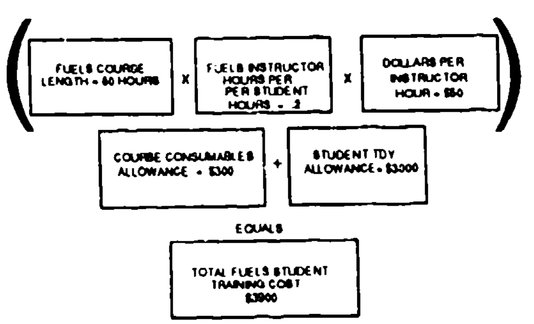


Figure 9 Example CER Process Table for Fuel System Technician Training Cost

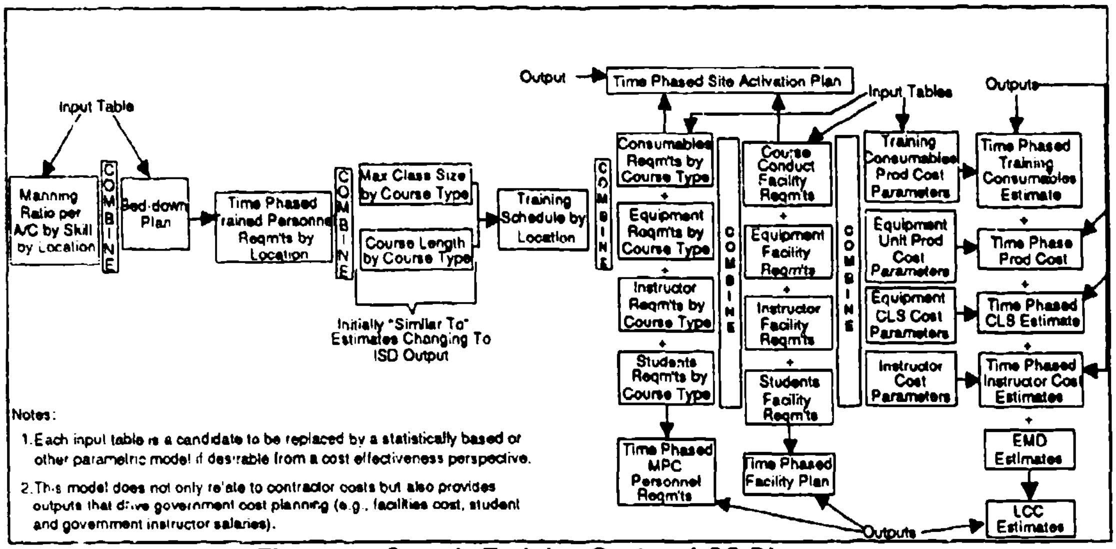


Figure 10 Sample Training System LCC Diagram

Figure 10 shows a view of what an overall model might entail and provides some sense of the complexity that may be involved. However, we do not require the use of a particular model or automated tool. Rather, we emphasis the thoughtful involvement of people in deciding each element of the Training System and discreetly identifying reasonable cost implications for each element. If automation helps the analysis or makes it less expensive to accomplish, use it. However, the automation should implement and clarify, rather than obscure, the dynamic relationships between a specific Training System configuration and LCC.

HOW MUCH ANALYSIS CAN WE AFFORD?

The length and intensity of the process can be influenced to a significant degree by limiting the depth to which input elements are determined and the amount of detail contained in the individual cost estimates and CERs. This normally carries a certain penalty in the inherent accuracy in the LCC determination, but is sometimes necessary in the interest of expediency or analytical economy. Simplified analyses are often appropriate early in a program when detailed knowledge of prime system characteristics is sketchy. It makes little sense to analyze training requirements below the level at which comparable prime system requirements are known. Eventually however, when a great deal of information regarding the training system configuration is available, it normally should be used to make the LCC determination. This will increase the accuracy of the estimates and our confidence level in them.

HOW DO WE DEAL WITH UNCERTAINTY IN THE ANSWERS?

Uncertainty is always present in all plans or analyses. In the future, prime items may be used differently; technological capabilities may change, and we may understand more about human learning behavior. These are a few of the factors that may place unanticipated demands on the Training System. The methodology that is presented in this paper is intended to deal only with discrete estimates for a specific system design. No inherent provisions are made for variability in estimates to cover uncertainty. This is a deliberate decision to simplify the analysis task and to make it more usable for a training specialist as opposed to a method that can only be practiced efficiently by analytical specialists.

Since some degree of uncertainty is always present, we need some method of addressing it. Two basic approaches can be applied that deal with any reasonable type of uncertainty. First, if the primary concern is the validity of the individual training element cost estimates,

creating pessimistic, optimistic and most likely estimates will be adequate to bracket the uncertainty associated with the LCC of a configuration. Secondly, if the uncertainty is more associated with the system configuration, alternative configurations can be created that establish high, medium and low complexity alternatives. Alternative configurations and alternative element cost estimates can be created to deal with both types of uncertainty. The range of possible LCC answers is then established by using repetitive runs of the LCC model for the bracket points that we have just established.

Dealing with alternative system configurations and uncertainty in estimating can be time consuming and expensive. The variability that can or should be addressed must be determined by the Training System development team. Their decision must be based upon the economic factors associated with doing the analysis, schedule considerations, and the criticality of the decisions that will be based on the analysis. The level of detailed knowledge that is available to develop hypothetical system descriptions and estimates also constrains our analytical options. Additionally, one can limit the analytical process quite legitimately by ensuring that only system configurations that meet all anticipated training requirements are subjected to detailed LCC analysis. It makes no sense to estimate the cost of a system if it has no prospect of meeting the anticipated performance requirements.

HOW DO WE USE THE DATA AFTER WE HAVE IT?

The objective in any LCC program is to first understand, then control, the costs associated with a product. In the case of Training System design, we can use the methods outlined in this paper to understand what influences Training System cost; then we can apply management and design action to control cost. One of the most significant benefits of these methods is that they force an understanding of a complete system configuration. Of equal value is the fact that decision makers can adjust the overall configuration of the Training System to optimize LCC performance. The current business emphasis on affordability in defense related products demands that we have an understanding of the complete cost picture regarding our products and that we do what is necessary to ensure that affordability.

CONCLUSIONS

Traditional LCC modeling techniques have all too frequently been viewed as something of a cultist practice and a black art. LCC estimates have often been viewed as having questionable accuracy and utility because the methods we used were largely practiced by analytical specialists, and we were never quite sure what the analysis tools were doing with the numbers. The methods in this paper force us to create orderly system configuration descriptions, ensure that we know what is in our LCC model, and make the logic associated with the analytical process obvious and repeatable. The approach is intuitive and work-man like. It is designed for use by training specialists who do not necessarily have an extensive background in LCC techniques. The structure of the process bases the analysis on a detailed system configuration description. This ensures that we have a sound basis to glue together all elements of our Training System. We then establish a rigorous, visible cost estimate that addresses all LCC implications of the system. The cost estimate is then readily traceable to the system configuration, employment concept and support plans that drive it.

GUIDELINES FOR CMI INTEROPERABILITY: THE AVIATION INDUSTRY STEPS FORWARD

Jack Q. Hyde
Boeing Commercial Airplane Group
Seattle, Washington

Anne Montgomery
Independent Computer Systems Consultant
Las Cruces, New Mexico

ABSTRACT

One of the most recent actions of the Aviation Industry CBT (Computer-Based Training) Committee (AICC) was to publish guidelines for the interoperability of Computer Managed Instruction. This paper describes the AICC guidelines for interoperability of CMI systems. It addresses

- How CMI systems in general function
- The value of interoperability
- Achieving interoperability: An overview of guidelines in three areas
    1) CMI/CBT interoperability: How different CMI and CBT systems from different vendors can work together.
    2) CMI/CMI interoperability: How different CMI systems can pass course structure and student management rules to other CMI systems.
    3) Lesson evaluation tools: How different data analysis tools can work with CBT from different vendors.

ABOUT THE AUTHORS

Jack Hyde has been working in Flight and Maintenance Training at Boeing for over 20 years. He started as a classroom instructor and training developer in the flight training ground school. In 1977, he designed and implemented his first CBT lesson using the PLATO system. Since then he as worked with WICAT and Authorware CBT systems as well. Currently he is working in a group called Customer Training Technology with a job title of Computer Technology Analyst.

Ms. Montgomery has over 20 years experience in the design, development, and implementation of Computer-Based Training (CBT) Systems, Computer-Managed Instructional (CMI) Systems, and Training Management Systems (TMS). She has developed customized CMI systems for both commercial and military training applications. In her role as technical coordinator of the Aviation Industry CBT Committee, she has worked with airframe manufacturers, CBT vendors, airlines, and CBT developers in the definition of CMI interoperability guidelines. Ms Montgomery is an independent consultant residing in Las Cruces, New Mexico.

GUIDELINES FOR CMI INTEROPERABILITY: THE AVIATION INDUSTRY STEPS FORWARD

Jack Q. Hyde
Boeing Commercial Airplane Group
Seattle, Washington

Anne Montgomery
Independent Computer Systems Consultant
Las Cruces, New Mexico

THE AICC

The Aviation Industry CBT (Computer-Based Training) Committee (AICC) is a 5-year old consortium of international CBT professionals. Membership includes the major airframe manufacturers and their suppliers, leading aviation-industry CBT vendors, airlines, and other standards-making and regulatory agencies. Its primary purpose is to generate guidelines which promote the economic and effective use of CBT within the aviation industry. In pursuit of this purpose, the AICC has published guidelines for the purchas of CBT delivery stations, standards for di d audio, and recommendations for selecting an operating system.

After conducting an airline survey to determine needs and desires regarding Computer-Managed Instruction (CMI), the AICC has recently completed the task of creating CMI guidelines. The goal of the guidelines is three-fold:

1) To allow each CMI system to be able to manage a variety of CBT lessons from different vendors, and
2) To enable CBT lessons to operate with a variety of CMI systems and data analysis tools, and
3) To allow course structure and student management rules to be passed from one CMI system to another

CMI OVERVIEW

CMI stands for Computer Managed Instruction. A CMI system is more than a scheduler of CBT materials. CMI systems are capable of managing both online (CBT) and offline instructional activities and tests. In general a CMI system has one or more of the following five components:

1. A component used for the development of course structures.

2. A testing component used for the development and administration of offline and online tests. Testing can be handled via
   - The CMI system
   - A separate test system (off line)
   - Traditional CBT.

   Each of these must be able to report test results to the CMI system.

3. A student rostering component enables entering student names and demographic data. Students may be grouped into classes with this component.

4. A component which provides student assignment management or routing including:
   - Administrator/instructor functions to oversee the day-to-day training operations and intervene when necessary
   - Assignment manager functions to control student assignments based on sets of rules (both predetermined and user-defined)
   - Standard approach to lesson initiation to provide a method for the CMI system to start-up lessons from different CBT vendors
   - Student logon functions to control and manage student access, maintain student-accessible data records, and display the student's current assignment

5. A component which provides student data collection and management including:
   - Functions to collect and maintain performance data on students at all levels of courseware presentation
   - Functions to provide standard analyses and outputs on performance data collected

INTEROPERABILITY OVERVIEW

In the past, authoring systems made the customer (the CBT administrator or user) a captive of the authoring system vendor. If the customer wanted to take advantage of CMI features in his courses, he had two choices.

1) Design his own CMI system with his authoring system tools, or
2) Purchase a CMI system from the same vendor who supplied the authoring system.

In either case, the resulting CMI system works only for a single vendor's CBT lessons. This is fine, until the customer aquires CBT courseware designed with a different authoring system, from a different vendor.

Several circumstances can motivate a customer to use CBT courseware incompatible with his CMI system.

- A manufacturer delivers incompatible courseware with a new airplane purchase.
- An airline purchases courseware from a vendor that uses a different authoring system.
- A customer decides to design new CBT lessons with a different authoring system.

There are many reasons a customer may wish to continue to use a single CMI system instead of multiple systems (different CMI systems for different groups of CBT lessons)

- Instructors are already familiar with a CMI system, and training on a new system would take time. This impacts the speed with which new courseware can be used, and the cost of training how to use it.
- It is desirable to maintain the student's overall "look and feel" in the airline's courseware. (The CMI/student interface provides a significant part of a course's look and feel.)
- Maintenence of two different CMI systems is more complex than maintaining a single system.
- The current CMI system has features and functions not available with the CMI associated with the new courseware.
- There is a desire to add some new lessons designed with a different authoring system to an existing course. A single CMI system is desirable for the entire course.

This paper describes the three aspects of CMI interoperability covered by the AICC guidelines; and suggests reasons why these aspects of interoperability are desirable.

The three aspects of interoperability discussed are:

- CMI management of CBT lessons.
- Moving course structure between systems.
- Storing lesson evaluation data.

CMI Management of CBT

There are two aspects of the AICC approach to enabling interoperability of CMI systems with different CBT systems.

1) **Lesson launch:** The CMI should have a standard approach to CBT lesson initiation, and
2) **Communication:** The CMI should have a standard approach to providing information and instructions to the CBT lessons, and receiving information from the CBT lessons. The AICC Guidelines define two files to enable this communication:
   - CMI to CBT: Lesson start-up information.
   - CBT to CMI: Information required by the CMI system to record student performance and perform the next lesson routing or assignment.

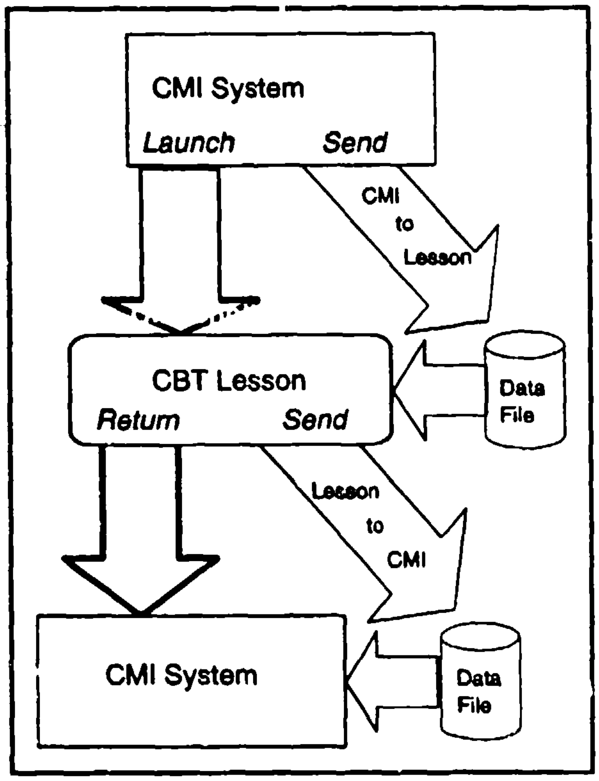


Figure 1: CMI Management of CBT

Process Summary -- Essentially this is how the interoperability works:

1. The CMI system creates a file containing the data necessary to start-up a CBT lesson. The file is created just prior to the initiation of the CBT system.

   The name of the CMI-to-CBT lesson data file must be known to the CBT application.

2. Once the CBT lesson is initiated, it reads the data file created by the CMI system and then deletes it. (Some lessons may not need this input file simply because student information is not necessary for the lesson.)

3. The CBT system must create a file containing data to be passed back to CMI so that the CMI system can update its student performance data and make the next assignment (perform routing activity).

   The CMI system passes in the file name for the lesson-to-CMI data file as part of the CMI-to-lesson core data.

4. When the student leaves the lesson, the CBT system updates and completes the file of information for the CMI system.

5. The CMI system reads the CBT-to-CMI data file, and using the information updates applicable student data kept by the CMI system and determines the next student assignment or routing activity.

   It is the responsibility of the CMI system to delete the CBT-to-CMI data file either immediately after determining the student's next assignment/routing activity or in such a manner as to insure that the disk space is managed properly and that there is no leftover data confusing the lesson.

Moving Courses

A course may be as simple as a few lessons to be viewed sequentially, or it may be as complex as hundreds of lessons, some of which are prerequisites to others and some of which may be experienced in any order. Basically, courses have two components: instructional elements and structure.

The instructional elements are all the lessons, tests, and other assignable units in the course. Frequently, the content elements also include all of the objectives to be mastered in the course.

In defining a structure, the developer frequently groups lessons for assignment. In other cases the designer defines complex lesson hierarchies. The AICC Guidelines accommodate both of these needs with the concept of a block. Blocks are simply groupings of instructional elements or other blocks.

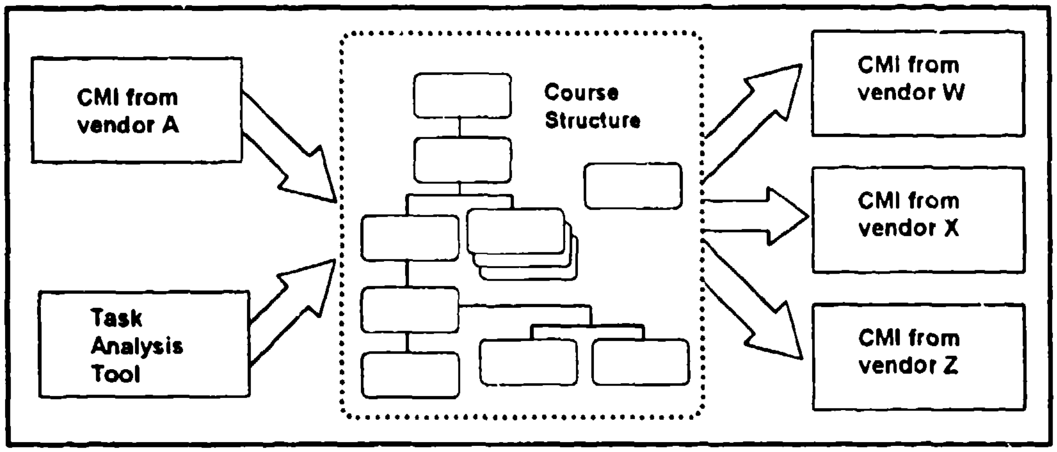


Figure 2: Moving Courses

The structure determines the order in which these are to be experienced by each student. The order may be quite complex, depending on prerequisites, or even student performance. The part of the CMI system that sequences the course content, is referred to as the *router*.

There are at least two circumstances in which guidelines for moving courses from one environment to another are useful. The first assumes a course is complete and is being transferred from a vendor or manufacturer to an airline -- moving from one CMI system to another. The second assumes a course is being designed in a tool other than a CMI system -- moving course design into CMI.

Transferring a new course into the existing CMI system manually, requires typing hundreds of lesson names, and duplicating all of the sequencing information. This requires a significant number of man hours. Having a standardized mechanism for describing course content and structure, enables CMI systems to "ingest" a new course with minimal manual effort.

There are many tools, other than a CMI system, which may be used to design a new course. One of the most common is a Task Analysis tool. If a course design tool can output a standardized description of a course, the CMI system can pull in the new course from that description. This can save hundreds of man hours of retyping and inputting data.

Storing Lesson Evaluation Data

Lesson evaluation data includes information that a CBT lesson or test generates on the behavior of a student (i.e. his performance. It may include such items as a student's responses, latency, and path through a lesson.

Lesson evaluation data can be used for

- Student performance analysis. Data collection of the student's interaction with the lesson. This helps to determine what the student knows, and what he learns. Comparing individual student progress with his peers gives a measurement of individual rate of learning.
- Item analysis. This can indicate how well an element of instruction trains; or how well a test question measures student performance. This enables quality control of the testing and instruction.
- Attitude survey. The determination of how well the student likes the courseware. How well the student feels the courseware is working. This aids in measuring customer satisfaction.
- Path optimization. The determination of the best sequencing of lessons and tests for a specific student. The determination of what material may be skipped by a student. The determination of what supplementary material or remediation is required by a student.

Standardizing the format of the student records permits multiple tools to use the information. By having standard interchange

formats, the market for analysis tools becomes much larger than just a single vendor's customers. Vendors are therefore encouraged to create sophisticated, easy-to-use analysis tools because of the payback of a larger customer base.

INTEROPERABILITY KEY: THE FILE

CMI and CBT systems must be able to communicate with each other in order to work together. Communication is essentially a flow of data from one program to another, or from one system to another.

The three data flows required for interoperability discussed in this document are:

- CMI ←→ CBT
- CMI → CMI
- CMI → Lesson-evaluation

In each of these cases the data flow can be handled with files. By creating guidelines for file format and content, the data can be understood by any CMI or CBT system.

The AICC selected two file formats for the data in these flows -- both are ASCII formats that are readable with any simple text editor:

- Microsoft Windows .INI file
- Comma delimited text file.

MS Windows INI Files

This file structure is based on the Microsoft WINDOWS *.INI files. The INI file contains three types of data -- **group, keyword**, and **comment**. The structure of the file and these data types are discussed below.

Groups provide a mechanism for dividing a file into manageable segments that are more easily accessed by data retrieval routines. They also provide a means to organize a file of data into logically related parts. This is helpful for human-processing of a file as well as computer processing.

Groups tend to be large data items, generally several lines in length. A group extends from its group identifier to the next group identifier, and may include multiple lines. Although groups may contain keywords, they may not contain other groups.

Keywords are names of data items that are limited in size to a single line. This generally limits the data to 60 or 70 characters. The data items associated with a keyword are referred to as keyword arguments or keyword values.

Comments are text that is of use to a human viewing a file. They are ignored by a computer processing the data in the file.

Table 1: INI File Elements

Appearance in file	Element name
[group]	Group
keyword=parameter	Valid Keyword
; groups and keywords ; may have comments	Comment

Example -- This file was created by a Lesson to pass information to a CMI system.

Table 2: Example Windows INI File

```
[CORE]
LESSON_STATUS = Passed
LESSON_LOCATION = End
SCORE = 87
TIME = 00:25:30
; this is the core group of data
; this is the lesson performance data for
; a passed lesson that
; required a time of 25 minutes,
; 30 seconds.
; The student recieved a raw score of 87
```

Comma Delimited ASCII Files

Data stored in a comma delimited ASCII file can be imported easily into virtually any off-the-shelf database product or spreadsheet. Many programs use this format to exchange data.

This format is not the same as a text file that is saved in ASCII form. Comma delimited format supplies a simple mechanism for separating records and fields and for distinguishing data types.

The record is the data found on a single line.

The field is the data that is found between commas (comma delimited) on the line. There is no fixed length for each field, and there is no fixed length for the records in the file.

Notice in Table 3 below, there are labels for each column. Each column corresponds to a field. Each row in the table corresponds to a record. In the conversion of this table to a comma-delimited file, the name of each field is gone. Only the field data itself is in the file. Position therefore becomes critical in determining the meaning of a field.

Notice also that empty fields, or blank fields may have to exist in the comma delimited file because the information is position dependent. In the third record there are two blank fields. The first is an empty number field, and the second is an empty text field.

Table 3: Example Table

Assignable Unit ID	Title	Type	Max Score	Duration	File Name
777APU-1	Auxiliary Power Unit	Tutorial	38	00:18:00	APU.EXE
777EL-1	Electrical Power, Part 1	Tutorial	41	00:23:00	ELEC1.EXE
777EL-2	Electrical Power, Part 2	Practice			ELEC2.EXE

Table 4: Comma Delimited File with Same Contents

```
"777APU-1","Auxiliary Power Unit","Tutorial",38,"00:18:00","APU.EXE"
"777EL-1","Electrical Power, Part 1","Tutorial",41,"00:23:00","ELEC1.EXE"
"777EL-2","Electrical Power, Part 2","Practice",,"","ELEC2.EXE"
```

CMI/CBT-LESSON COMMUNICATION

CMI and Lesson communication is two way. The CMI system sends information to the lesson when it begins. The lesson sends information to the CMI system when the lesson ends.

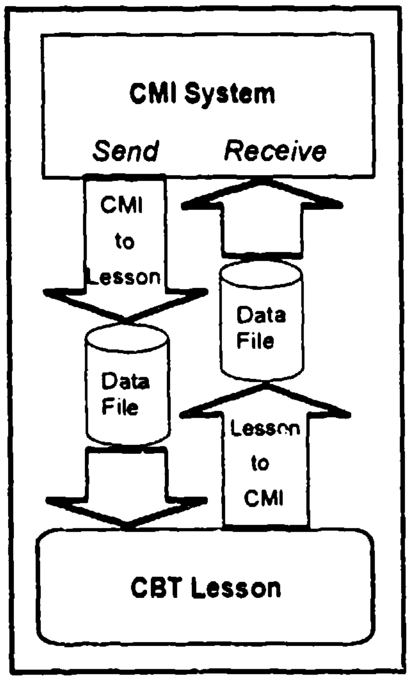


Figure 3: Communication Data Files

The information is sent in a file -- two files actually. The first file is created by the CMI system, and the second is created by the lesson.

CMI to CBT File

This is information that a typical lesson obtains from a CMI system to enable it to perform the functions expected of it. In Table 5, core items are listed first, followed by the optional group names alphabetically. (In the file, group names may be in any order.) After each group name are the keywords (if any) which are appropriate for that group. (In the file, keywords may appear in any order inside their group.)

A core item is one which must always be provided by the CMI system to be AICC compliant. Core items are those which a lesson may always depend upon being available. The lesson may or may not use the core items, but they are available if required.

Optional items are group and keyword data which may be needed by a lesson to perform optimally. However, the lesson must be constructed such that there is a default to be used if these optional items are not provided by the CMI system.

Table 5: CMI to CBT File Contents

Group Names and Keywords	Function of Group
[Core] Student_ID Lesson_location Student_Name Lesson_Status Output_File Score Lesson_Mode Time	Information required to be furnished by all CMI systems. What all lessons may depend upon at start up, from any AICC compliant CMI system.
[Core_Lesson] data is undefined and may be unique to each lesson	Information held by the CMI system for the lesson since last student attempt.
[Core_Vendor] data is undefined and may be unique to each vendor	Required information for some lessons. Must be furnished by CMI system.
[Comments] no key words <delimited>	E-Mail type information that an instructor or administrator wants to send to a student.
[Evaluation] Course_ID Comments_file Interactions_file Objectives_status_file Path_file Performance_file	File names and locations where the lesson should store the lesson evaluation information.
[Objectives_Status] J_ID.01 J_Score.01 Local_ID.01 J_Status.01	Information on each objective in an assignable unit.
[Private_Area] Path	Area where lesson can find and or store lesson-unique data.
[Student_Data] Attempt_Number Cumulative_Time Mastery_Score Max_Time_Allowed Time_Limit_Action Lesson_Status.01	Information on student performance expectations.
[Student_Demographics] Age Job_Title Birth_Date Language City Native_Language Class Race Company Religion Country Sex Experience State Familiar_Name Street_Address Instructor_Name Telephone Years_Experience	Personal information on student. Characteristics relating to student before course entry.
[Student_Preferences] Audio Text_Color Bookmarks Text_Location Lesson_Type Text_Size Text Video Window.01	Student selected options collected in previous lessons, or previous instances of this lesson.

CBT Lesson to CMI File

This is information that a lesson must/may make available to a CMI system. The core items (which the lesson MUST make available) are first, followed by the optional items listed alphabetically. Starting this file should be the first thing done by the lesson after launch

Table 6: Lesson to CMI File Contents

Group Names and Keywords	Function of Group
[Core] Lesson_Location Lesson_Status Score Time	Information required by the CMI system to function.
[Core_Lesson] data is undefined and may be unique to each lesson	Information required by the lesson. Passed to the CMI system to hold and return at the next start-up
[Comments] no key words <delimited>	Student comments on lesson.
[Objectives_Status] J_ID.01 Local_ID.01 J_Score.01 J_Status.01	Information on objectives contained in the lesson.
[Student_Preferences] Audio Bookmarks Lesson_Type Text Text_Color Text_Location Text_Size Video Window.01	Student selected options to be passed to next lesson he enters.

COURSE STRUCTURE DATA

The purpose of defining a CMI structure interchange format, is to simplify the process of moving a course from one system to another.

After moving a course, a review-and-modify effort is going to be required. The existence of standard interchange files however, should eliminate a large number of the manhours necessary to input a new course from scratch.

Basic Concepts

The files containing the structure of a course need to answer the question, "What information does a CMI system need, to present the training material to the student in the way desired by the designer?"

The approach taken by AICC guidelines assumes that the answer can be *implied* in a table that contains all of the lessons and lesson groups in a course.

The answer can be made *explicit* by stating prerequisites for each lesson (or assignable unit) in the course. When pre-conditions are set that must be met before a student can select or be assigned a lesson, each lesson, assumes a place in the course structure.

For instance, assume there is a course of six lessons. The order of the lessons can be implied by putting them in a simple table (Table 7); then reading the table left to right, and top to bottom.

To make this order explicit, assume lesson 6 has a prerequisite of the student having completed lesson 5, and lesson 5 requires passing lesson 4, and lesson 4 requires completion of lesson 3, etc. (Shown in Figure 4) This results in the linear presentation of the lessons in sequence from 1 through 6.

Table 7: Course Hierarchy Table

Root	Lesson 1	Lesson 2	Lesson 3	Lesson 4	Lesson 5	Lesson 6

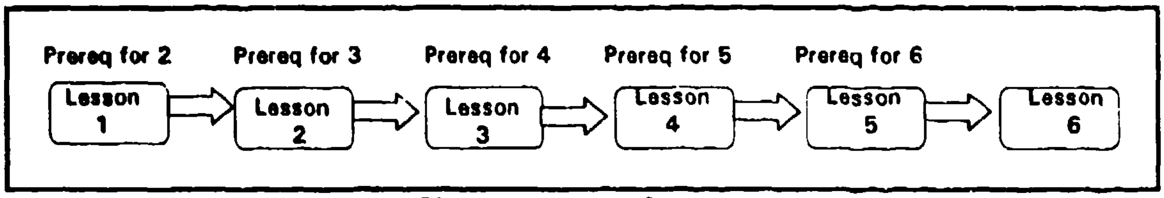


Figure 4: A Simple Course

In the AICC approach, prerequisites can be defined in terms of completed lessons, or mastered objectives. Table 8 reflects the prerequisites shown in Figure 4.

Table 8: Prerequisite Table

Assignable Unit	Prerequisites
Lesson 1	None
Lesson 2	Lesson 1
Lesson 3	Lesson 2
Lesson 4	Lesson 3
Lesson 5	Lesson 4
Lesson 6	Lesson 5

Of course, even with prerequisites there are cases where it is desirable to let the student chose the order in which he attempts some lessons. If three lessons have exactly the same prerequisites, then the student has an option -- after meeting the prerequisites -- of selecting any of the three.

In addition to files describing the course hierarchy and prerequisites there need to be files describing the elements in the course. This is textual information and not required to determine the order in which the student can take the course material. This information includes the titles of the various items in the course and a narrative description of them when desired.

Levels of Complexity

The AICC guidelines define five levels of complexity in describing the course structure. Increasing the level of complexity from level 1 to 2 to 3 to 4 to 5 should result in:

- Less effort to review and modify the CMI system after importing the data.
- More complete description of the designer's intended usage of the course material.

There are up to seven files that can be used to describe a course's content and structure. The level of complexity determines the number of files required and the amount of information required in each file. The following sections briefly describe the contents or purpose of each file. Tables 9 through 17 identify the names of each field, keyword or group in each of the seven possible files.

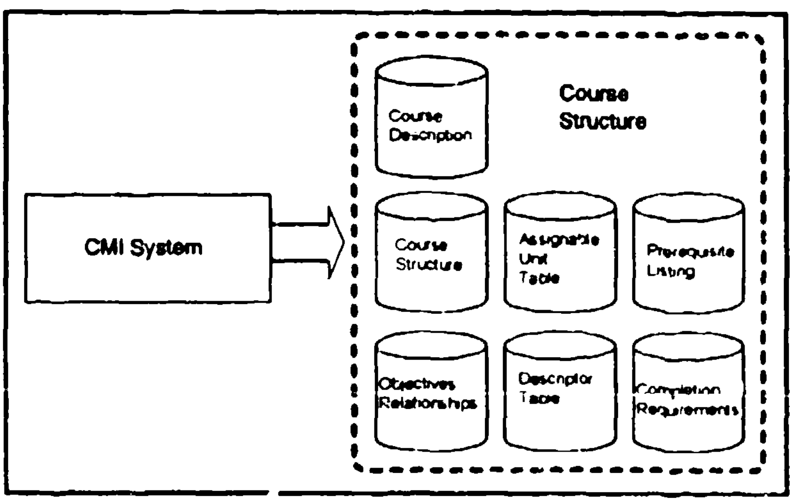


Figure 5: Course Structure Files

Course Description File (Table 9)

This file contains information about the course as a whole. It contains a textual description of the course, and general makeup of the course -- the number and type of elements.

Table 9: Course Description File

Groups and Keywords	
[Course]	
Course_ID	Total_AUs
Course_Title	Total_Blocks
Level	Total_Objectives
Course_Creator	Total_Complex_Obj
Course_System	Max_Fields_CST
	Max_Fields_ORT
[Course_Description]	

Assignable Unit Table (Table 10)

This file contains information about the assignable units (AUs) in the course. Each assignable unit has its own record (or row in the table). The information includes the name of the AU, its ID, and the mastery score for that assignable unit.

Descriptor Table (Table 11)

This file contains a complete list of every course element in the course. Course elements include:

- Assignable Units
- Blocks
- Objectives
- Complex Objectives

It is used as the basic cross reference file showing the correspondence of system-generated IDs with user-defined IDs for every element. This file also contains any textual description created for an element in the course.

Course Structure Table (Table 12)

This file contains the basic data on the structure of the course. It includes all of the assignable units and blocks in the course, showing how they are organized -- which AUs are members of which blocks. And finally, it implies the order in which these should be taken.

Table 10: Assignable Unit Table -- the fields

System ID	Type	Command Line	Duration	File Name	Max Score	Mastery Score	System Vendor	Core Vendor

Table 11: Descriptor File -- the fields

System ID (for course element)	Developer ID (for course element)	Title	Line number	Description

Table 12: Course Structure Table

Block	Members -- Assignable units & other blocks			
Root	System ID	System ID	System ID	
System ID	System ID	System ID	System ID	System ID
System ID	System ID	System ID		

Table 13: Objectives Relationships Table

Structure Element	Members: Assignable units, blocks, & objectives			
System ID	System ID	System ID	System ID	System ID
System ID	System ID	System ID		
System ID	System ID	System ID	System ID	
System ID	System ID	System ID	System ID	System ID
System ID	System ID	System ID		

Objectives Relationships File (Table 13)

Objectives have complex and variable relationships to other elements of a course. For instance, a lesson may cover several objectives or a single objective may require mastery of several lessons. Other objectives may require the mastery of many sub-objectives.

The Objectives Relationships file is able to define all of these relationships. This file is optional, depending on the level of the course description.

Prerequisite Listing (Tables 14, 15, and 16)

Sometimes it may be desirable to prevent a student from entering a lesson until he has met certain prerequisites. This file allows that sort of constraint to be placed on each block or assignable unit (AU) in a course.

There are three levels of complexity that may be used in describing prerequisites. The first (Table 14) allows a single prerequisite AU or block to be defined for each element in the course. The second (Table 15), allows prerequisites to be defined in the form of a logic statement (with "ands" and "ors") that includes objectives. The third (Table 16), and most complex prerequisite listing allows the definition of prerequisites for each mode in which the lesson may be used. Possible modes are:

- Review
- Browse
- Normal

Level 2

Table 14: Prerequisite Listing

Structure Element (Block or AU)	Prerequisite (Block or AU)
System ID	System ID
System ID	System ID
System ID	System ID

Level 3, 4

Table 15: Prerequisite Listing – More Complex

Structure Element (Block or AU)	Prerequisite Logic Statement (Blk, AU or Obj)
System ID	System ID & System ID
System ID	System ID | System ID
System ID	System ID
System ID	System ID & (System ID | System ID)

Level 5

Table 16: Prerequisite Listing – Most Comprehensive

Structure Element (Block or AU)	Prerequisite Logic Statement (Blk, AU or Obj)	Mode
System ID	System ID & System ID	N
System ID	System ID | System ID	B
System ID	System ID	R
System ID	System ID & (System ID | System ID)	N

Level 5

Table 17: Completion Requirements File

Block or Complex Objective	Completion Logic Statement (Blk, AU or Obj)
System ID	System ID & System ID
System ID	2*(System ID | System ID | System ID)
System ID	System ID
System ID	System ID & (System ID | System ID)

Completion Requirements (Table 17)

While lesson and objective status is determined within the lesson by the logic designed into it, this is not true of blocks. Blocks are created specifically to describe a course structure. Similarly Complex Objectives are defined in terms of other structure elements. Therefore, block and complex objective status must be determined by the CMI system.

The Completion Requirements file is designed to allow the explicit specification of when a block or objective is complete when it does not conform to the defaults for completion. It is essentially an exception file.

Lesson evaluation data is contained in several files. The file names for this data are passed to the lesson from the CMI system. If the file already exists, the lesson appends the data. If the file does not exist, the file is created and the data deposited.

With lesson evaluation data, analysis tools and CMI systems are able to assemble information on multiple lessons, multiple uses of the same lesson, and multiple students.

The analysis of the information is not the subject of these guidelines. What is covered here is essentially raw data.

All of these files are optional. Up to five of them may be required to store all of the information desired from a CBT lesson.

LESSON EVALUATION DATA

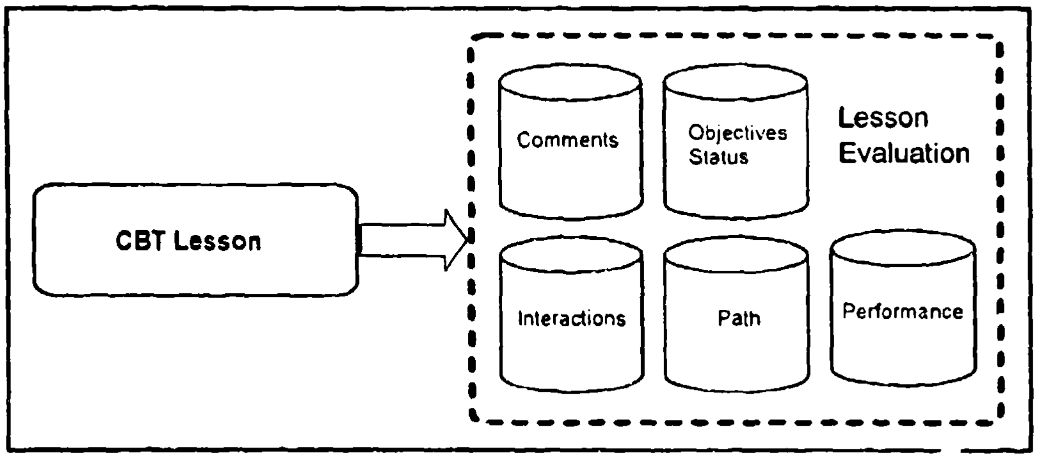


Figure 6: Lesson Evaluation Files

Table 18: Comments File -- the fields

Course ID	Lesson ID	Date	Time	Location	Line Number	Comment

Comments File (Table 18)

This is a journal file that contains freeform feedback from the student. It contains his criticisms and complements -- recorded as he moves through the lesson. It is a duplicate of the [Comments] group that is passed to the CMI system in the CBT-to-CMI file.

Interactions File (Table 19)

In this context, an interaction is a recognized and recordable input or group of inputs from the student to the computer. All of the items in this file are .elated to a recognized and recorded input from the student (or lesson user.)

Most commonly, these interaction records will be student responses to questions. The types of questions with defined response types are:

- True/False
- Multiple choice
- Fill in the blank
- Matching
- Simple performance
- Sequencing
- Likert
- Unique

Objectives Status (Table 20)

This file contains comprehensive information on objectives, including their ID and their status (passed, failed, or not attempted.)

Table 19: Interactions File -- the fields

Course ID	Lesson ID	Date	Time	Interaction ID	Objective ID	Type interaction	Correct response	Student response	Result	Weighting	Latency

Table 20: Objectives Status -- the fields

Course ID	Lesson ID	Date	Time	Objective ID	Local ID	Score	Status	Mastery time

Path File (Table 21)

This file allows an analysis of what path the student took through a lesson. It enables the analyst to determine when the student asked for help, when he selected alternative branches, if he selected optional instruction, and the order in which he proceeded through the lesson.

Table 21: Path File – the fields

Course IDs	Lesson ID	Date	Time	Element Location	Status	Why Left	Time in Element

ADDITIONAL INFORMATION

To obtain the complete documentation of the AICC standard, or to get more information on the AICC, contact:

Scott Bergstrom
AICC Administrator
University of North Dakota
Box 8216, University Station
Grand Forks, ND 58202-8216

Telephone: (701) 777-4380

GLOSSARY

argument Keyword argument. The information relating to a keyword that appears to the right of the equal sign. Also called keyword value or keyword data.

assignable unit The smallest element of instruction or testing to which a student may be routed by a CMI system. It is the smallest unit the CMI system assigns and tracks.

A program or lesson launched by the CMI system.

AU Abbrieviation for "assignable unit."

block An arbitrarily defined grouping of course components. Blocks are composed of related assignable units or other blocks.

This is a term used in the AICC document CMI Guidelines for Interoperability. A block may correspond to any level of the AICC instructional hierarchy above lesson, up to and including course.

bookmark Identification of a location in a lesson to which a student plans to return. Bookmarks are placed by the student for his own reference and review purposes.

CAI Computer-aided Instruction. Sometimes Computer-assisted Instruction. Normally used as a synonym for CBT. However some make the following distinction between CAI and CBT.

CAI (cont.) CAI: The computer as an aid to learning. Supports instruction, but is not the prime medium for delivery of instruction. Uses include presentation or practice but not both.

CBT: Computer as the primary mode of instruction.

CBT Computer-Based Training. The use of computers to provide an interactive instructional experience. Also referred to as CAI (Computer Assisted Instruction), CAL (Computer-aided Learning), CBE (Computer Based Education), CBI (Computer-based Instruction), etc.

CMI Computer-Managed Instruction has several definitions. In its broadest sense, it includes the following:

1) Rostering and storing student information.

2) Scheduling students and resources.

3) Computer acquisition and storage of student performance data. This is frequently referred to as student data collection instead of CMI.

4) Data presentation. After the data has been collected, it can be massaged by the computer, providing meaningful summaries for human interpretation. This is frequently referred to as data analysis instead of CMI.

CMI (cont.)

5) And finally, the computer can make decisions based on its analysis of the student's performance. It can manage the student's learning. It makes decisions as to what material the student should cover next, what material is not necessary, and what remedial actions if any, should be taken.

In some contexts, the term CMI excludes data collection and data analysis. The strictest definition of CMI includes only the fifth aspect, the computer management of the student.

The combination of items 3) and 4) above, is frequently referred to as "Student Evaluation."

core item

Data in a file for CMI/Lesson communication. A core item is one which must always be provided to be AICC compliant. Core items are those which a lesson may always depend upon being available. The lesson may or may not use the core items, but they are available if required. Most core items are in a single group entitled "core" (or "CORE" or "Core").

course

A complete unit of training. A course generally represents what a student needs to know in order to perform a set of related skills or master a related body of knowledge.

course (cont.)

Level 2 in the AICC Hierarchy of CBT Components:

1. Curriculum
2. Course
3. Chapter
4. Subchapter
5. Module
6. Lesson
7. Topic
8. Sequence
9. Frame
10. Object

course elements

Three items which constitute the building blocks for a course description. Each of these building blocks has its own title and attributes.

- Assignable Unit (lesson)
- Block, and
- Objective.

curriculum

A grouping of related courses.

Level 1 in the AICC Hierarchy of CBT Components:

1. Curriculum
2. Course
3. Chapter
4. Subchapter
5. Module
6. Lesson
7. Topic
8. Sequence
9. Frame
10. Object

demographics

Information associated with a student prior to entering a course. Student attributes. Typical demographic data includes the student's name, age, sex, years of experience, and native language.

group A unit of information in a standardized file for storing CMI information. Groups are large data items, generally several lines in length. A group extends from the group identifier to the next group identifier, and may include multiple lines. All carraige returns and symbols between group identifiers may be significant, depending on the definition of the specific group. Although groups may contain keywords, they may not contain other groups.

hierarchy The structure of lessons and/or courses which, to a large extent, determines how the student will perceive the course organization and in what order his lessons will be assigned.

interaction An exchange between a student and a program, beginning with a screen touch, a mouse click, a keyboard, or other input by a student, followed by an on-screen reaction of the program.

In the context of the CMI guideline for storing student performance data: A recognized and recordable input or group of inputs from the student to the computer.

item analysis. This can indicate how well an element of instruction trains; or how well a test question measures student performance. This enables quality control of the testing and instruction.

lesson A meaningful division of learning that is accomplished by a student in a continuous effort -- that is at one sitting. That part of the learning that is between designed breaks. Frequently requires approximately 20 minutes to an hour. OR

A grouping of instruction that is controlled by a single executable computer program.
Or
A unit of training that is a logical division of a subchapter, chapter, or course.

Level 6 in the AICC Hierarchy of CBT Components:

1. Curriculum
2. Course
3. Chapter
4. Subchapter
5. Module
6. Lesson
7. Topic
8. Sequence
9. Frame
10. Object

lesson element An arbitrary division of an assignable unit that has been uniquely named (has its own ID). An assignable unit may have from two to hundreds of lesson elements.

keyword A unit of information in a standardized file for storing CMI information. Keywords are names of data items that are limited in size to a single line. This generally limits the data to 60 or 70 characters.

performance analysis — Determination of a student's capabilities, based upon data collection of the student's interactions within one or more lessons. This helps to determine what the student knows, and what he learns. Comparing individual student progress with his peers gives a measurement of individual rate of learning.

router — Software which sequences a series of lessons, tests, and other assignable units in a course. The router determines the order in which the student experiences segments of his computer-based training.

structure elements — The parts of a course which can be uniquely assigned by a CMI system. These are units that can be rearranged to determine the order in which a student experiences a course of instruction. There are two structure elements in the AICC view of a course description:

- Assignable unit (the lesson)
- Block

value — Keyword data. The information relating to a keyword that appears to the right of the equal sign. Also called keyword argument.

TOWARD ASSESSING TEAM TACTICAL DECISION MAKING UNDER STRESS: THE DEVELOPMENT OF A METHODOLOGY FOR STRUCTURING TEAM TRAINING SCENARIOS

Joan K. Hall
Daniel J. Dwyer
Janis A. Cannon-Bowers
Eduardo Salas
and
Catherine E. Volpe

Naval Training Systems Center
Orlando, Florida

ABSTRACT

Tactical decision making teams in the modern warfare environment are faced with situations characterized by rapidly unfolding events, multiple plausible hypotheses, high information ambiguity, severe time pressure, and severe consequences for errors. Training interventions should fully exploit instructional designs that will enable teams to maintain performance under these stressful conditions. Recent research indicates that training scenarios should incorporate significant task situations (events) that present opportunities to learn and achieve desired performance requirements. In addition, the event-based approach allows for standardized, reliable, and valid measurement of team member performance. However, little guidance exists regarding how training scenarios should be designed so that they will have a significant impact on helping the team maintain performance under stressful conditions. Therefore, the purpose of this paper is threefold. First, a stress assessment methodology (SAM) will be described that guide in the creation of structured training scenarios so that they contain appropriate and relevant levels of situational stressors. The SAM is based on the idea that training scenario design should be driven by an identified standard of performance. Therefore, two evaluation instruments will be described, the Behavior Observation Booklet (BOB) and the Sequenced Actions and Latencies Index (SALI), whereby an assessment of team member performance is obtained at pre-specified, time-tagged events in the training scenario. Lastly, implications for creating event-based training scenarios are discussed.

ABOUT THE AUTHORS

Joan K. Hall is a Research Psychologist in the Human Systems Integration Division of the Naval Training Systems Center (NTSC). She is responsible for conducting behavioral research to develop training and simulation principles for the Tactical Decision Making Under Stress (TADMUS) exploratory research program. She received her Ph.D. in Industrial/Organizational Psychology from the University of South Florida. Her research interests are in the areas of training for stress exposure, tactical decision making, and the development of training and simulation principles to mitigate the effects of stress on performance.

Daniel J. Dwyer is a Research Psychologist in the Human Systems Integration Division of the Naval Training Systems Center. His experience is in the areas of tactical decision making under stress, embedded training, computer-assisted instruction, maintenance training, performance measurement, and training effectiveness evaluations. He holds an M.S. degree in Industrial/Organization Psychology from the University of Central Florida, and is currently a doctoral candidate at the University of South Florida.

Janis A. Cannon-Bowers is a Research Psychologist in the Human Systems Integration Division of the Naval Training Systems Center. She is co-principal investigator for the TADMUS program, and is responsible for behavioral research concerned with improving individual and team decision making in the Navy tactical environment. She received her Ph.D. in Industrial/Organization Psychology from the University of South Florida.

Eduardo Salas is a Senior Research Psychologist in the Human Systems Integration Division of the Naval Training Systems Center. He is the principal investigator for NTSC's behavioral research program on team training and performance. He is currently the project manager for the TADMUS and Aircrew Coordination Training efforts. He received his Ph.D in Industrial/Organizational Psychology from Old Dominion University.

Catherine E. Volpe is a Post Doctoral Fellow in the Human Systems Integration Division of the Naval Training Systems Center. She is responsible for conducting behavioral research in support of NTSC's project on Tactical Decision Making Under Stress (TADMUS). Her research interests are in the areas of inter-positional clarification training, tactical decision making, and the development of training and simulation principles to mitigate the effects of stress on performance.

TOWARD ASSESSING TEAM TACTICAL DECISION MAKING UNDER STRESS: THE DEVELOPMENT OF A METHODOLOGY FOR STRUCTURING TEAM TRAINING SCENARIOS

Joan K. Hall
Daniel J. Dwyer
Janis A. Cannon-Bowers
Eduardo Salas
and
Catherine E. Volpe

Naval Training Systems Center
Orlando, Florida

INTRODUCTION

Tactical decision making teams in the modern warfare environment are faced with scenarios characterized by rapidly unfolding events, multiple plausible hypotheses, high information ambiguity, severe time pressure, sustained operations, and severe consequences for errors (Cannon-Bowers, Salas, & Grossman, 1991). In order to adapt to these stressors, team members must learn to coordinate their actions so that they can gather, process, integrate, and communicate information in a timely and effective manner. Therefore, training interventions should fully exploit instructional designs that will enable teams to maintain performance under stressful conditions.

Recently, a number of team research programs have resulted in guidelines and promising strategies for team training design and evaluation. For example, Cannon-Bowers et al. (1991) have made recommendations regarding the systematic development of training for enhancing the tactical decision making of anti-air warfare (AAW) Combat Information Center (CIC) teams under stress. Hall, Driskell, Salas, and Cannon-Bowers (1992) have provided guidelines for developing stress exposure training. Prince, Oser, Salas, & Woodruff (1993) have proposed augmented guidelines for simulator scenario development used in crew resource management (CRM). Fowlkes, Lane, Salas, Oser, and Prince (1992) have described a methodology for developing measures of observable team performance during CRM. Baker and Salas (1992) have provided principles for measuring teamwork skills. Dwyer (1992) has developed an index of team performance that is based on observable indicators of effective and ineffective behaviors across several critical functions of an AAW team.

Taken together, these researchers suggest that team training requires the development of structured scenarios which provide the opportunity to perform critical team actions (Prince et al., 1993). A key factor in the design of training scenarios is determining how to structure events which elicit appropriate decision making actions. A realistic range of operational conditions and stressors should be inserted in the scenarios so that a representative sample of decision making actions can be observed and measured (Hall et al., 1992). Therefore, evaluating the efficacy of team training objectives requires that scenarios be designed so that team performance can be measured in a standardized, relevant, valid, and reliable manner. Currently, however, very little guidance exists regarding how training scenarios should be constructed so that they will have a significant impact on helping a team maintain tactical decision making performance under stressful conditions.

In response to this issue, the purpose of this paper is threefold. First, the development of a stress assessment methodology (SAM) will be described that guides in the creation of structured training scenarios so that they contain appropriate and relevant levels of situational stressors. The SAM is based on the idea that training scenario design should be driven by an identified standard of performance. Therefore, two evaluation

instruments will be described, the Behavior Observation Booklet (BOB) and the Sequenced Actions and Latencies Index (SALI), whereby an assessment of team member performance is obtained at pre-specified, time-tagged events in the training scenario. Thirdly, implications for creating event-based training scenarios are discussed.

THE SAM

For purposes of illustrating the development of the SAM, the BOB, and the SALI, initial findings from a Navy research program called Tactical Decision Making Under Stress (TADMUS) will be described (Cannon-Bowers et al., 1991). TADMUS was initiated to address the problem of maintaining individual and team decision making in the CIC environment by applying recent advances in decision theory, training, and display design (Cannon-Bowers et al., 1991). A major goal of the TADMUS program is to conduct research to understand how combat-related stress affects tactical decision making of AAW CIC ship teams (e.g., tactical action officer, identification supervisor, anti-air warfare coordinator, tactical information coordinator, and electronic warfare supervisor). One of the tasks of this program has been to develop AAW research scenarios with appropriate levels of stressors, and with the capability for evaluating team performance. Development of the SAM was required to accomplish this task. Following is a description of the steps involved in the initial development of the SAM which included: (a) identification of relevant task stressors, (b) specification of significant AAW scenario events that represent task stressors, and development of an event-based scenario, and (c) documentation of specific scenario features to demonstrate different levels of task stressors. Development of the BOB and SALI will be described later in this paper.

Identification of Relevant Stressors

Fortunately, the potential for developing team training scenarios for complex simulation exercises has been advanced in recent years with the development of such process-tracing techniques as concept mapping, protocol analysis, cognitive task analysis, the critical decision method, and Cognitive Network of Tasks (COGNET) (Zachary, Zaklad, Hicinbothom, Ryder, Purcell, & Wherry, 1991). These techniques enable identification of key decision making tasks, and the associated knowledge, skills, and abilities required to perform the tasks (Klein, 1993; Redding, Cannon, Lierman, Ryder, Purcell, & Seamster, 1991; Rouse & Valusek, 1993; Woods, 1993; Zachary et al., 1991).

We utilized several of these techniques as a first step toward incorporating appropriate AAW stressors into the TADMUS scenarios. First, surveys were conducted with CIC personnel that were specifically involved in the AAW area. They indicated that large numbers of commercial and military air traffic (heavy workload), and conflicting or missing information (information ambiguity) about the air traffic were highly relevant stressors in the AAW area.

Secondly, results of work by Zachary et al. (1991), using the COGNET strategy, determined the knowledge requirements of a key member of the AAW CIC team, the AAW Coordinator (AAWC). Essentially, the two major AAW tasks involve threat response management and situation assessment. Threat response management includes evaluating the threat status of aircraft, and planning strategy and tactics (i.e. preplanned responses). Situation assessment includes understanding the geo-political and the tactical picture, the ship's resource status, and ship team relationships (e.g., the battle group, ownship, and AAW team). This analysis confirmed that effective AAW tactical decision making required the operator to process large amounts of tactical data in a short period of time, and to "deconflict" the information made available to the operator (Zachary et al., 1991).

Scenario Development

Once workload and information ambiguity had been identified as stressors, we had four Navy Subject Matter Experts (SMEs) incorporate them into a "moderately" stressful, and a "highly" stressful 30 minute AAW scenario. So that each AAW team member would experience stress, SMEs were asked to create fictional scenarios with events that were likely to require action by all the team members. The

scenarios do not contain events that have actually occurred. Both scenarios involve an AAW CIC ship team located in the Northern Persian Gulf. The ship's mission is to monitor the movement of military and commercial air traffic. Both Scenarios A and B retained the same event structure (10 events each) and timing of events so that team member performance could be compared, but Scenario B had more aircraft and ambiguous events incorporated into it. Table 1 shows the basic event structure shared in Scenarios A and B. The first event occurs at the very beginning of the scenario (time zero). The last event, J, occurs at almost 28 minutes.

Table 1.
<u>The Basic Event Structure Shared in Scenarios A and B.</u>

EVENT	EVENT TIME MINUTES + SECONDS	EVENT DESCRIPTION
A	00 + 00	FRIENDLY CAP APPEARS, UNDER FRIENDLY AWACS CONTROL.
B	10 + 30	POSSIBLE HOSTILE F-4s, MULTIPLE BOGIES, DETECTED BY OWN SHIP, B-105, R-124NM, C-232, S-365KTs, A-9500FT, CLIMBING.
C	12 + 00	FRIENDLY CAP, DIRECTED BY AWACS, INITIATES INTERCEPT VECTOR TO THE NE, S-380KTS.
D	14 + 30	FOUR POSSIBLE HOSTILE BOGIES APPEAR TO SPLIT INTO TWO SECTIONS, SLIGHTLY DIVERGING IN COURSE, B-112,R-111NM, COURSES 230 TO 235, S-365 KTS, A-12KFT.
E	17 + 00	ALL BOGIES FEET WET, B-122, 104NM; FRIENDLY CAP, 40NM SW OF BOGIES, CONTINUES TO CLOSE FOR INTERCEPT.
F	19 + 30	APQ-120 INTERMITTENTLY DETECTED TO THE SOUTHEAST FROM THE HIGH F-4s.
G	21 + 42	CAP INTERCEPTS BOGIES, CONFIRMS TWO POSSIBLE HOSTILE F-4s, B-123, R-79NM, C-305, S-365KTS, ALTITUDE 10000FT.
H	23 + 30	TWO HI POSSIBLE HOSTILE F-4s, WITH CAP IN COMPANY TURN SOUTHEAST, B-122, R-67NM
I	24 + 30	UNIDENT (F-4D) TURNS TO 010 FOR APPROACH LEG TO KHARK ISLAND; APQ-120 LOST, B-053, WHEN ACFT TURNS AWAY
J	27 + 30	TWO POP-UP RADAR CONTACTS B-123, R-46NM, SECOND SECTION OF POSSIBLE HOSTILE BOGIES, C-305, S-365KTS, A-500FT.

The scenarios were then keyed into a simulation facility, composed of five PCs networked with a file server workstation, that had been configured to support AAW tactical decision making research (Holl & Cooke, 1989).

Documenting Stressors

The next step in the SAM process was to document the level of workload and ambiguity in both Scenarios A and B. The workload assessment matrix and the ambiguity assessment matrix were created to be used as tools to evaluate scenario stress levels, as well as to create future scenarios.

Workload Assessment Matrix. Essentially, the total number of air targets to be correctly prosecuted in a 30-minute AAW scenario is a workload indicator. However, results of the cognitive task analysis by Zachary et al. (1991) provided a way to obtain a more accurate representation of the degree to which each target creates work for the team. They found that a main goal of an AAW operator is to evaluate the threat status of air traffic. In order to do this, the operator manages workload by mentally placing the aircraft or "track" of interest into at least one of the following four activity categories: (a) unknown track, (b) interest track, (c) action track, and (d) engageable track. The findings by Zachary et al. (1991) suggested that each category requires an increasing level of operator activity in order to evaluate the aircraft. Although the degree of increased workload has not been determined, the distinction between categories enables us to, at the very least, identify task features that may hinder optimal task performance.

Following is a description of each category. An unknown track requires minimal mental activity because no actions have been taken to identify this track, and it is designated as workload level 1. If a track is designated an interest track, this means it must be monitored, because it is a potential threat, or is a friendly track that the team must be aware of for coordination in an engagement. This designation indicates a higher level of workload than an unknown track, and is labeled as workload level 2. A target is designated as an action track if it requires some action to be taken, besides monitoring (e.g., warnings, issue report, request for information). This designation indicates a higher level of workload than an interest track, and is labeled as workload level 3. A target is designated as an engageable track if it meets the rules of engagement, and indicates a higher level of workload than an action track. An engageable track is designated as workload level 4.

The workload assessment matrix was designed to evaluate scenario tracks in terms of these four categories. To illustrate, we evaluated event J from Scenarios A and B with the workload assessment matrix. Table 2 shows the workload assessment matrix with workload activity levels for air tracks during Event J of Scenario A. Five possible hostile F4s fit category 3 as action tracks. The SMEs had incorporated these tracks into the scenario for the specific purpose of creating an opportunity for the AAW team to perform a variety of the behaviors (e.g., issuing warnings, reports, and requesting information from team members). Four friendly commercial aircraft, two commercial helicopters, and three friendly military aircraft were listed as interest tracks. The Navy SMEs had added these aircraft to Scenario A to create some monitoring activities for the AAW team. The total number of action and interest tracks in event J in Scenario A were five and nine, respectively.

Table 2.
Workload assessment matrix with workload activity levels for air tracks during Event J of Scenario A.

TRACK	UNKNOWN 1	INTEREST 2	ACTION 3	ENGAGEABLE 4
Possible Hostile F4 #1			•	
Possible Hostile F4 #2			•	
Possible Hostile F4 #3			•	
Possible Hostile F4 #4			•	
Possible Hostile F4 #5			•	
Commercial Aircraft #1		•		
Commercial Aircraft #2		•		
Commercial Aircraft #3		•		
Commercial Aircraft #4		•		
Commercial Helicopter #1		•		
Commercial Helo #2		•		
Friendly AWACS		•		
Friendly Tanker		•		
Friendly CAP		•		

Table 3 shows the workload assessment matrix with examples of workload activity levels for air tracks during the same event for Scenario B. The total number of interest and actions tracks in event J for Scenario B was 12 and eight, respectively. It could be assumed that Scenario B requires more mental workload during Event J than Scenario A.

This type of evaluation could be carried out for all the events in the scenario to obtain a more complete estimate of mental workload. In addition, the workload assessment matrix could be applied to surface (ships) and subsurface (submarines) tracks.

Table 3.
Workload assessment matrix with workload activity levels for air tracks during Event J of Scenario B.

TRACK	UNKNOWN 1	INTEREST 2	ACTION 3	ENGAGEABLE 4
Possible Hostile F4 #1			•	
Possible Hostile F4 #2			•	
Possible Hostile F4 #3			•	
Possible Hostile F4 #4			•	
Possible Hostile F4 #5			•	
Possible Hostile F4 #6			•	
Possible Hostile F4 #7			•	
Possible Hostile P3C			•	
Commercial Aircraft #1		•		
Commercial Aircraft #2		•		
Commercial Aircraft #3		•		
Commercial Aircraft #4		•		
Commercial Aircraft #5		•		
Commercial Aircraft #6		•		
Commercial Helicopter #1		•		
Commercial Helo #2		•		
Friendly AWACS		•		
Friendly Tanker		•		
Friendly EP3E Aircraft		•		
Friendly CAP		•		

<u>Ambiguity Assessment Matrix</u>. The ambiguity assessment matrix was used to evaluate each critical event in a scenario in terms of the amount of vague, conflicting, and missing information occurring for air traffic. Ambiguous information about a target should lead to greater workload on the AAW team because they must actively pursue information about the track in order to identify it. To illustrate, we evaluated Scenarios A and B with the ambiguity assessment matrix. Table 4 shows the ambiguity assessment matrix with two examples of vague, conflicting, or missing information during Scenario A. For example, during Event B, at time 12 minutes and 30 seconds, electronic emissions from a possible hostile F4 are lost from the ship's radar. At this point, the ship's team must increase its monitoring and action activities to determine what happened to this track.

Table 4.
The ambiguity assessment matrix with examples of vague, conflicting, or missing information in Scenario A.

EVENT/ TIME	TRACK(S)	VAGUE, CONFLICTING, OR MISSING INFORMATION
B/ 12+30	1 Possible Hostile F4	Electronic Emissions Lost From Ship's Radar
C/ 10+30	1 Possible Hostile F4 Detected on Radar	Intelligence Reports Received by Ship Indicated Multiple Possible Hostile F4s Departing Shiraz

Table 5 shows the ambiguity assessment matrix with three examples of ambiguous information for Scenario B. For example, during Event I, at time 27 minutes, an internal report is received by the ship's team indicating a possible floating mine close to the ship. In this instance, the team must begin actions to validate this report. As with the workload assessment matrix, the ambiguity assessment matrix could be applied to identifying vague information regarding surface and subsurface tracks, as well.

Summary of SAM

Initial development of the workload assessment matrix and the ambiguity assessment matrix has shown evidence of their utility for evaluating stressors in AAW research scenarios. Furthermore, they can be used to specify and build new scenario features. However, in order to further validate this methodology, a measurement system that ties scenario events to team performance is required. Below is a description of a methodology for measuring event-based team performance that is being used in the TADMUS project to compare and evaluate team member responses to Scenarios A and B.

Table 5.
The ambiguity assessment matrix with examples of vague, conflicting, or missing information in Scenario B.

EVENT/ TIME	TRACK(S)	VAGUE, CONFLICTING, OR MISSING INFORMATION
A/ 3+30	1 Possible Hostile F4	Electronic Emissions Lost from Ship's Radar
I/ 27+00	Internal Report	Possible Floating Mine Close to Ship
J/ 30+30	1 Unidentified Possible Hostile F4s joins with a Section of 2 other Unidentified Possible Hostile F4s, and are headed toward ownship	Unclear Determination of Aircraft Intent

A METHODOLOGY FOR MEASURING EVENT-BASED TEAM PERFORMANCE

The accurate diagnosis of performance shortfalls and the tailoring of subsequent training toward correcting these shortfalls for teams and team members is contingent upon systematic performance assessment (Fowlkes et al., 1992). One of the benefits of employing the SAM approach is that it allows for an event-based scenario structure which can serve as the basis for performance measurement. The AAW scenarios created for TADMUS can be described as a set of critical events that serve as a basis for developing tactical decision making performance objectives. Consequently, the Navy SMEs that helped create Scenarios A and B were also enlisted to develop performance standards for each of five AAW team members: tactical action officer, identification supervisor, anti-air warfare coordinator, tactical information coordinator, and electronic warfare supervisor. They identified critical observable behaviors for each team member that should occur at specified events in Scenario A.

Behavior Observation Booklet

Next, the BOB was developed for each AAW team member position. Figure 1 is an example of a page from the BOB for AAWC actions during Event J of Scenario A. Seven actions were identified that the AAWC can be expected to take for event J. Upon observing the AAWC's performance in response to a pop-up radar contact on two aircraft, trained observers rate the individual's overall performance quality on a five-point scale, ranging from 1 (poor) to 5 (very good). Space is made available to add actions. This evaluation is carried out for each event, and for each team member participating in the scenario. For each team member, the BOB scores are averaged across all of the events in the scenario to derive an overall BOB score for that individual.

CIC TEAM POSITION: ANTI-AIR WARFARE COORDINATOR
EVENT J: POP-UP RADAR CONTACT, SECOND SECTION
TIME: 27 MINUTES AND 30 SECONDS

STEP 1

1. Looking for Aircraft profile
2. Issue external report
3. Issue new track number
4. ____________

STEP 2

1. Ensure Tactical Action Coordinator issues trip wire warning calls
2. Recommend enhanced alert posture/equipment
3. Configure ship's position to TAO
4. Direct/modify that team configure equipment accordingly
5. ____________

OVERALL PERFORMANCE QUALITY FOR EVENT J

1	2	3	4	5
VERY POOR	POOR	AVERAGE	GOOD	VERY GOOD

OVERALL SEQUENCING QUALITY FOR EVENT J

1	2	3	4	5
VERY POOR	POOR	AVERAGE	GOOD	VERY GOOD

Figure 1 Example of page from BOB (Performance Quality) and SALi (Sequencing Quality) for AAWC actions during Event J of Scenario A.

Sequenced Actions and Latencies Index

Also, Figure 1 shows an example of the two-step sequence in which an AAWC is supposed to perform the seven actions. The SALI is an overall quality determination of whether the AAWC performed the actions in or out of the sequence shown. Upon observing the AAWC's performance in response to a pop-up radar contact on two aircraft, trained observers rate the individual's overall sequence quality on a five-point scale, ranging from 1 (poor) to 5 (very good). For each team member, the SALI scores are averaged across all of the events in the scenario to derive an overall SALI score for that individual.

Summary of the BOB and SALI

The advantage of the BOB and SALI is that they can be used as a diagnostic tool for observing and evaluating team member performance over the course of a scenario run. A major advantage to this measurement system is that immediate feedback to team members can be provided to improve performance. Results of team member performance can be charted against other team members on a timeline to determine areas of performance that require improvement. The key to the BOB and SALI is that they document observable team member actions. While the development and use of these measures may be somewhat labor intensive, the complexity of team performance requires observation (Baker & Salas, 1992). The technology for enabling observers to capture information about team behaviors needs improvement.

IMPLICATIONS FOR CREATING EVENT-BASED TRAINING SCENARIOS

Ultimately, the main objective of scenario development is to provide an opportunity for team members to perform critical behaviors (Fowlkes et al., 1992). The SAM is based on the idea that training scenario design should be driven by an identified standard of performance. Once performance objectives have been defined, training scenarios should be built so that they present opportunities to learn and achieve the desired performance requirements. SAM addresses the issue of identifying, assessing, and manipulating appropriate levels of situational stressors to create a productive learning context, and to enable effective assessment of performance objectives using the BOB and SALI. One of the main lessons learned in the initial development of the SAM is that scenario development and identification of performance standards is an iterative process. Navy SMEs provided input about changes in Scenario A and B events that were necessary to ensure behaviors were elicited from all the AAW team members.

This is the first step in providing guidelines for designing scenarios for team training. Future work in this area will include refinement and validation of the SAM, the BOB, and the SALI, use of the SAM to produce scenarios with different levels of stressors, and application of SAM, BOB, and SALI to other training situations. This same methodology could be transferred to other types of stressors and task situations (e.g., army battleforces, air traffic control, and nuclear power plant operations). In conclusion, the following are recommendations that are offered as a point-of-departure in the design of training scenarios and the measurement of team performance.

- To identify stressors, employ process tracing techniques that will enable detection of key complex decision making tasks, and the knowledge, skills, and abilities to perform the tasks.

- Use the workload assessment matrix and the ambiguity assessment matrix to assemble scenario events. In this way, different levels and types of task features can be systematically incorporated into a series of training scenarios.

- Utilize SMEs to help incorporate stress events into scenarios that will elicit key observable decision making actions.

- To create a BOB and SALI, utilize SMEs to identify key team member actions and action sequences that should occur at each significant event in the training scenario.

- Keep in mind that development of training scenarios and performance

measurement instruments should be closely tied together, and changes in performance objectives should influence training design.

REFERENCES

Cannon-Bowers, J. A., Salas, E., & Grossman, J. D. (1991, June). Improving tactical decision making under stress: Research directions and applied implications. Paper presented at the International Applied Military Psychology Symposium, Stockholm, Sweden.

Baker, D. P., & Salas, E. (1992). Principles for measuring teamwork skills. Human Factors, 34, 469-475.

Dwyer, D. J. (1992). An index for measuring naval team performance. Proceedings of the Human Factors Society 36th Annual Meeting, 2, 1356-1360.

Fowlkes, J. E., Lane, N. E., Salas, E., Oser, R. L., & Prince, C. (1992). Targets for aircrew coordination training. Proceedings of the 14th Interservice/Industry Training Systems Conference (pp. 342-352). Washington, DC: National Security Industrial Association.

Hall, J. K., Driskell, J. E., Salas, E., & Cannon-Bowers, J. A. (1992). Development of instructional design guidelines for stress exposure training. Proceedings of the 14th Annual interservice/Industry Training Systems Conference (pp. 357-363). Washington, DC: National Security Industrial Association.

Holl, R. E. & Cooke, J. R. (1989). Rapid software development: A generic tactical simulator/trainer. Proceedings of the 11th Annual Interservice/Industry Training Systems Conference (pp. 337-342). Washington, DC: National Security Industrial Association.

Klein, G. A. (1993). A Recognition-primed decision (RPD) model of rapid decision making. In G. A. Klein, J. Orasanu, R. Calderwood, & C. E. Zsambok (Eds.), Decision making in action: Models and methods (pp. 138-147). New Jersey: Ablex.

Prince, C., Oser, R., Salas, E., & Woodruff, W. (1993). Increasing hits and reducing misses in CRM/LOS scenarios: Guidelines for simulator scenario development. The International Journal of Aviation Psychology, 3, 69-82.

Redding, R. E., Cannon, J. R., Lierman, B. C., Ryder, J. M., Purcell, J. A., Seamster, T. L. (1991). The analysis of expert performance in the redesign of the en route air traffic control curriculum. Proceedings of the Human Factors Society 35th Annual Meeting, 2, 1403-1407.

Rouse, W. B., & Valusek, J. (1993). Evolutionary design of systems to support decision making. In G. A. Klein, J. Orasanu, R. Calderwood, & C. E. Zsambok (Eds.), Decision making in action: Models and methods (pp. 270-286). New Jersey: Ablex.

Woods, D. D. (1993). Process-tracing methods for the study of cognition outside of the experimental psychology laboratory. In G. A. Klein, J. Orasanu, J., R. Calderwood, & C. E. Zsambok (Eds.), Decision making in action: Models and methods (pp. 228-251). New Jersey: Ablex.

Zachary, W. W., Zaklad, A. L., Hicinbothom, J. H., Ryder, J. M., Purcell, J. A., & Wherry, R. J. (1991). COGNET representation of tactical decision-making in ship-based anti-air warfare (NCCOSC technical report in preparation). San Diego, CA: Naval Command, Control and Ocean Surveillance Center.

THE EFFECTS OF ABOVE REAL-TIME TRAINING (ARTT) ON THREE TASKS IN AN F-16 PART-TASK SIMULATOR

Dutch Guckenberger
ECC International Corporation/UCF
Orlando, Florida

Kevin C. Uliano, Norman E. Lane & Kay Stanney
University of Central Florida
Orlando, Florida

ABSTRACT

In this application of ARTT, 24 mission-capable F-16 pilots performed three tasks on a part-task F-16A flight simulator under varying levels of time compression (i.e., 1.0x, 1.5x, 2.0x, and random). All subjects were then tested in a real-time (1.0x) environment. The three tasks under study were an emergency procedure (EP) task, a 1 versus 2 air combat maneuvering task, and a stern conversion or air intercept task. In the EP task, all ARTT pilots performed the EP task with 28% greater accuracy, and were better at dealing with a simultaneous MIG threat, reflected by a six-fold increase in the number of MIG kills compared to a real-time control group. In the ACM task, those pilots trained in the mixed time accelerations were faster to acquire lock, and were faster to kill both MIG threats than the other groups. In the stern conversion task, there were no statistical differences between groups.

These findings are generally consistent with previous findings that show positive effects of task variation (including time variations) during training. Results are discussed in the context of expansion and evolution of ARTT research across multiple simulator platforms and different types of high performance tasks. Also discussed are related research findings that support the benefits of ARTT. Further, a synthesis of mutli discipline research outlining the underlying theoretical basis for ARTT is presented. A proposed model of ARTT based on an analogy to Einstein's theory of special relativity is suggested. Conclusions and an outline of future research directions are presented.

Dutch Guckenberger received his M.S. in Simulation Engineering and his B.S. in Computer Science & Physics from the University of Central Florida (UCF). He currently teaches graduate courses in real-time simulation and is pursuing his Ph.D. in IEMS: Simulation Systems at UCF. Mr. Guckenberger is the Senior Software Design Engineer for ECC International Corporation's R&D Department whose specialties include real-time graphics, firmware and human sensory abilities. He has been involved with simulation for 11 years and has concentrated on ARTT research for the past 4 years. He is a member of IEEE, ACM, and SIGRAPH.

Kevin C. Uliano received his M.S. degree in Industrial Psychology and his B.A. in General Experimental Psychology from UCF. He has publications in the area of human factors psychology and is a member of the HFS and SIGCHI.

Dr. Norman E. Lane has nearly 30 years experience in human factors engineering, training, and personnel systems research and development. He has managed large scale Navy programs in areas such as simulator evaluation and utilization, training for altered perceptual environments, and performance measurement systems. Dr. Lane has more than 110 publications, from reference books to articles for general audiences. Dr. Lane is an internationally recognized expert in the area of skill acquisition and retention.

Dr. Kay Stanney, an assistant professor at UCF, is a graduate of the School of Industrial Engineering at Purdue University. She is currently developing a collaborative virtual reality (VR) at UCF, whose objective is to facilitate the wide-spread use of VR technology. Dr. Stanney's primary areas of research include human-computer interaction and cognitive engineering. She is a member of IIE and Human Factors Society.

THE EFFECTS OF ABOVE REAL-TIME TRAINING ON THREE TASKS IN AN F-16 PART-TASK SIMULATOR

Dutch Guckenberger
ECC International Corporation/UCF
Orlando, Florida

Kevin C. Uliano, Norman E. Lane & Kay Stanney
University of Central Florida
Orlando, Florida

INTRODUCTION

Above Real-Time Training (ARTT) refers to a training paradigm that places the operator in a simulated environment that functions at faster than normal time. In the case of air combat maneuvering, a successful tactical air intercept which might normally take five minutes, would be compressed into two or three minutes. All operations of the intercept would correspondingly be accelerated such as airspeed, turn and bank velocities, weapons flyout, and performance of the adversary. In the presence of these time constraints, the pilot would be required to perform the same mission tasks to the same performance criteria–as he would in a real time environment. Such a training paradigm represents a departure from the intuitive, but not often supported, feeling that the best practice is determined by the training environment with the highest fidelity. ARTT can be implemented economically on existing simulators. It is important to realize that ARTT applications require the simulated velocity of the targets and other entities to increase, NOT the update rate. Over 25 years ago NASA Dryden's flight test engineers recognized that if one could program a simulator to operate in "fast time", one could give test pilots a more accurate experience or "feel" of real-world stresses that would be present in the aircraft (Kolf 1973, Hoey 1976).

The origin of support for ARTT, in simulators, comes from anecdotal reports from NASA. Researchers at the NASA Dryden Flight Research Center during the X-15 program in the late 1960's needed a mechanism to address the X-15 test pilots' post flight comments of being "always behind the airplane..." and "... could never catch up" (Thompson, 1965). Clearly, there were some differences between the perceived time in the well-practiced simulator flights and perceived time in the experimental aircraft. The first time NASA used fast time simulation was toward the end of the X-15 program. Pilots compared practice runs at various time constants with flights they had already flown. A fast time constant of 1.5x felt closest to their flight experience and was planned on being implemented in the lifting body programs, but lack of funding precluded the program from fully developing the capability. Regardless, NASA's test pilots at DFRC have endorsed the use of "fast time" simulation as part of the training process (Kolf 1973, Hoey 1976).

Vidulich, Yeh, and Schneider (1983) examined the utility of time compression as a training aid for training a basic air traffic control skill (a high performance skill). One group practiced the intercept with the target plane traveling at 260 knots. The second group practiced the intercept at 5200 knots – 20 times real time! The subjects in this group received between 72-80 trials per hour during training. Both groups were then tested in real time. The time compressed group was significantly better at identifying the turn point; there was no difference between groups on estimating roll out heading for the intercept.

Guckenberger, Uliano, and Lane (1992), using a table top tank gunnery simulator, trained naive subjects on three tank gunnery scenarios under five acceleration factors (i.e., 1.0x, 1.5x, 2.0x, sequential, and mixed). Their results demonstrated that training time could be cut up to 50% with performance staying equal to or surpassing a real-time control group.

Further, in one ARTT group (mixed presentation) their mean performance scores were 50% higher than the control group (1.0X).

THEORETICAL UNDERPINNINGS

Psychophysical research into time perception has shown the relativistic nature of time perception in humans. (Jones 1976, Toumodge 1990, Skelly 1993). Relativistic nature of time is defined as linking a human observer's perception of time to that particular observer's "stimulation state" or "time norm"; analogous to Einstein's theory of special relativity linking relative velocities to a particular observer frame of reference norm. It is noteworthy that this analogy was arrived at independently by Jones (1976), Toumodge (1990) and Guckenberger (1992), from three different fields. Hahn and Jones (1981) have even developed working models though their work is primarily in the area of Audio training. Dr. June Skelly is attempting to extend the Audio finding to the arena of Visual training and has already generated some impressive initial results (Skelly 1993). This evolving multi-disciplinary research forms a firm theoretical basis upon which to build. The foundations for ARTT and Human perception are well established. Time perception can be altered if a particularly boring or interesting task is introduced, or if the arousal state of the subject is changed through external environmental cues (Parasuraman, 1986). Humans perceive time differently depending upon the individual's "stimulation state" or "time norm". This stimulation state is based, in part, on the sensory cues in the environment and the interactivity level between the individual and his/her environment. Perceived time, therefore, is tied to the particular individual at his/her particular stimulation state to form a "time frame of reference" for that individual. Cohen (1964) discusses evidence for an interrelationship between one's "inner clock" and sensory/motor functioning where each can influence each other to alter the perception of time. Most high performance tasks involve both sensory/motor and cognitive skills. Schnieder suggests a mild time stress to enhance high performance skills training (Schnieder 1985). Further Wright-Patterson Researchers have developed a method of Rapid Communication (RAP-COM) which improved throughput and retention (Matin & Boff 1988).

When this subjective time reference is perceived as long, it may offer a unique advantage for providing training on critical high performance skills. This artificially accelerated frame of reference may give the operator more "perceived time" in which to actually perform key elements of the mission. The very realization that the operator has more time may lead to better decision making and situational awareness. It may give the operator the edge that makes the difference in today's modern battlefield. It is important to note that when using ARTT more compressed training trials can be performed in the same amount of time. Even ignoring the performance gains, more training trials per unit time is reason enough to implement ARTT. As long as no negative training is introduced, more economic training can occur on existing simulators. The simplest case for ARTT is improved simulator usage either by more trials per unit time per trainee, or higher trainee throughput.

RESEARCH OBJECTIVES AND HYPOTHESES

The objectives of this task is to conduct research regarding: (1) the relative effectiveness of ARTT versus conventional training on different simulator platforms; (2) the relative effectiveness of alternative implementations of ARTT; and (3) the impact of ARTT versus conventional training on total time.

Prior research suggests that training in a time accelerated environment should lead to poor performance versus a control group, but should lead to greater performance on a real-time transfer task. Second, it is expected that there will be group differences in training as a function of the time acceleration constant that is used. Third, it is obvious that training time will be reduced in direct proportion to the time acceleration constant used. Finally, it is not expected that training under various time manipulations will lead to negative transfer of training to a real-time task.

METHOD

Subjects

Twenty-four mission-capable F-16 Air Force pilots from the 56th Tactical Training Wing, MacDill Air Force Base, Tampa served as subjects for this experiment. This subject pool had 743 mean flight hours (range of 300-3400), and 134 mean simulator hours (range of 30-500).

All subjects were recruited on a voluntary basis in accordance with American Psychological Association (APA) Principles for Research with Human Subjects.

Prior to testing, subjects were given written instructions informing them as to the general nature of the experiment.

Equipment and Materials

Two Avionics Situational Awareness Trainers (ASAT) were used as the testbed for this study. The ASAT is a low-cost F-16A cockpit trainer designed primarily to train in the beyond visual range (BVR) environment. The hardware components that make up the ASAT consist of three personal computers (PCs). The host computer is a PC-AT with an i386 CPU and a i387-20 co-processor, which drive the head-up (out-the-window) and radar electro-optic (REO) displays and collect the data coming from the stick and throttle. Another PC-AT computer (i286), drives the radar warning receiver display. Sound and vibrational cues are provided through the third PC which drives a stereo amplifier, seat and back cushion-mounted speakers, and sub woofers. Aural cues available in the ASAT include radar sensor tones, engine and air noise, missile launch, and gunfire, radar warning receiver (RWR) tones, and missile seeker head tones.

Graphics for the head-up display are high resolution, 1024 x 1024 RGB, with a 63.36 kHz horizontal scanning frequency. The monitor for the head-up and visual display is a 19-inch color CRT monitor which is mounted in front of the pilot on top of the cockpit enclosure, and gives the pilot a 23° X 23° field-of-view. The REO display simulates that of the F-16A Block 15S AN/APG 66 radar, and is presented on a 5" monochrome monitor. It is driven by the i386 and is controlled through switch activation on the throttle and by a radar control panel located on the left side of the simulator. The panel contains active switches to control antenna azimuth, antenna elevation and target history selection. The radar warning receiver (RWR) simulates the ALR-69 RWR, and the display consists of a 9" EGA resolution color monitor. All symbology is generic and unclassified.

The side-stick controller and throttle are high fidelity copies of the controls used in the actual F-16A. The stick can experience a maximum deflection of 0.25" in each of the four axis (forward, backward, right, left), and is equipped with buttons that allow the performance of different functions which include four way trim, missile release, gun triggering, missile select button (AIM 9-J/L), and a return to search switch. The throttle controls thrust from idle to full military power and beyond through five stages of afterburner. (It should be noted that no change in thrust results in the ASAT from afterburner stage 2 through stage 5; the afterburner has only two states: on and off.) Other throttle functions include: four way radar cursor, UHF/VHF transmit switch, missile uncage button, speed brake switch, antenna elevation knob, chaff/flare release button, and dog fight switch.

The ASATs communicate via a PC-based ethernet network at the asynchronous rate of approximately 10-14 packets per second. For the purpose of this experiment, the network was modified so that each ASAT communicated through a Hewlett-Packard i386, 33 mHz PC which served as the experimental interface. This PC controlled task selection, trial start and stop times, duration, data storage, and other experimental information. In this design, the PC would also send messages to either ASAT instructing the simulator to activate or deactivate certain functions (e.g., sound) that were required for a subject to perform a given task. Special purpose *C* and assembly software was written to handle these special requirements.

Procedure

The subjects' first mission was to familiarize themselves with the simulator, including its displays, controls, and handling qualities. These aspects of the simulator are probably different than what the subjects are normally accustomed to. Since the F-16A model is no longer in service with the U.S. Air Force, only some of our subjects had ever flown it. Based on preliminary test subjects, we do not believe this to be a problem since the F-16A and F-16C models have sufficiently similar aerodynamic and avionics characteristics. The subjects were given approximately forty-five minutes for familiarization across a wide variety of scenarios. During this time, the subjects were encouraged to test and experiment with the control and displays, and the flying characteristics of the simulator.

After the familiarization period there was about a fifteen minute break. The subjects then flew an assigned order of the three tasks at an assigned ARTT value.
These assignments had been made beforehand and represent a complete counterbalancing of the four ARTT conditions, three tasks, and 24 subjects.

For each task, the subject flew 10 trials at the assigned group, subjects were presented with a random presentation of the first three time constants. The within-group factor tested a trial effect with each subject receiving 10 training and 4 test trials (i.e., transfer of training trials). Dependent variables included varied flight performance data such as time-to-lock, time-to-kill, hit/miss percentage, mission performance times, and emergency procedure checklist performance. Specific data collected were a function of the task being performed

Training Tasks and Initial Conditions The three tasks used for this study are listed and explained below. A task ended when the subject "killed" the target(s) or when the task timed-out. We limited any given task to five minutes to optimize data storage. For each hop for each task, the subject had unlimited fuel. The subject did not have access to any ground control intercept (GCI) or airborne AWACS information. The following task briefings were the only information available.

> Task 1 - One versus Two Air Combat Maneuvering. Two bogeys on the nose at 25,000 ft. Goal was two valid face shots on the initial merge. Continue to engage the bogeys until they have been killed, or until the experimenter terminates the hop.
>
> Task 2 - Stern Conversion. Bogey was 40 miles on the nose at 20,000 ft. Goal was to perform stern conversion and position for a possible AIM 9J missile or gun shot as quickly as possible. Maximum distance for weapons employment was 1500 ft. The subject was required to maintain a 30 degree aspect cone at no more than 1500 feet before permission to fire was given. This allows for adequate data collection. This hop ended when the bogey has been killed or when the experimenter terminates the hop.
>
> Task 3 - Emergency Procedure. In this task, the subject was flying over enemy area suspected of having energy pulse weapons (better known as "power sucker"). The subject must deal with two external threats. Namely, the "power sucker" and an enemy bogey. When the subject was painted by one of these weapons, he heard (and felt) a constant low rumbling noise indicating an imminent and catastrophic power loss. If this happened, the emergency procedure (EP) to defeat this weapon was as follows:
>
> 1) fire energy decoy (missile);
> 2) change heading left 10 degrees;
> 3) hit energizer (flare);
> 4) change heading right 10 degrees;
> 5) fire energy decoy (missile);
> 6) hit energizer (flare).

If the subject performed the procedure above exactly, and in the correct order, the "power sucker" would be defeated and aircraft power would be restored. If not, the subject would crash. The goal of this task was to perform the EP above as quickly as possible while at the same time successfully engaging a hostile bogey.

RESULTS

Raw flight performance data originally collected at a 10-14 Hz iteration rate were reduced into trial summaries. Summary data were then analyzed using the Statistical Package for the Social Sciences (SPSS) (SPSS, 1992). The multivariate analysis of variance (MANOVA) syntax for SPSS was used as the overall design structure for the analysis; however, univariate F tests were calculated for specific planned comparisons of interest. These planned comparisons focused on identifying statistically-reliable differences between the performance of the four time acceleration groups in training, and performance comparing the average of the three training blocks (for a given task/dependent variable combinations) with the two transfer trial blocks.

For the emergency procedure (EP) task, number of MiGs killed, time to complete EP, and percent of EP performed correctly were analyzed by group. Analysis of the EP flight data demonstrated a significant increase in MIG kills from training to transfer for all accelerated conditions ($F_{3,20}$ = 10.87, $p<.01$) with the 1.5x and 2.0x conditions slightly outperforming the mixed group. The three accelerated groups, at the conclusion of the last transfer block, had a better than six-fold advantage in the number of MIG kills compared to those trained at real-time (see Figure 1.) Further, the groups accuracy for the EP at real-time was: Mixed-> 100% accurate, 2.0x-> 96.6% accurate, 1.5x-> 90% accurate, 1.0-> 72% accurate

When comparing performance in training on the number of MIG kills, there is also a significant difference between the groups ($F_{3,20}$ = 3.95, $p <$ .05). Both the 1.5x and 2.0x groups performed better in training when compared to the 1.0x and mixed groups. This finding was not expected, and is not consistent with what is known about the contextual interference phenomenon.

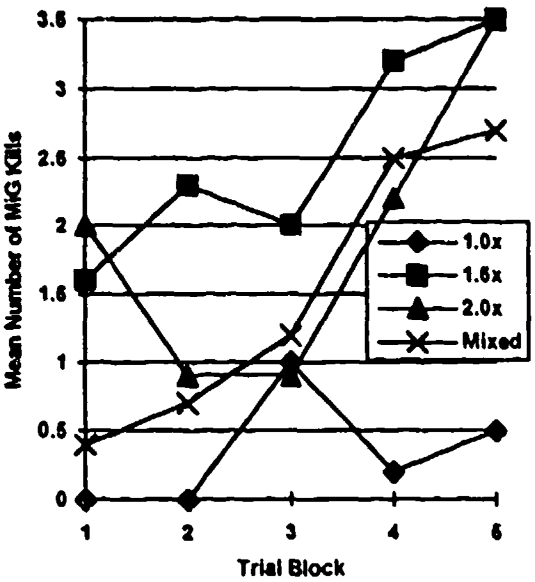


Figure 1. Mean Number of MIG Kills by Trial Block

Next, the time to complete the EP procedure, and percent of EP procedure performed correct were analyzed. As time went on, all the groups completed the EP checklist items quicker, although that difference was not statistically reliable. When comparing the accuracy performance, however, both the 2.0x and mixed conditions performed the checklist task significantly better than either the 1.0x or 1.5x groups, when later tested at real-time ($F_{3,20}$ = 7.45, $p <$ .002). In fact, subjects in the mixed group scored perfectly in the transfer condition. The 1.0x and 1.5x groups actually saw a slight decrease in accuracy performance from training to transfer. There were no mentionable differences between the groups in training.

For the stern conversion task, time to reach criterion, stern score, and distance at lock were analyzed by group.
Analysis of the stern conversion task showed that the 1.5x group performed only slightly better than the other groups in the time to reach a preset position criterion. The 1.5x group performed the task faster in training *and* in transfer but the reader will note that these findings are not statistically significant.

For the distance at lock variable, which represents a measure of radar target acquisition performance, the 2.0x and 1.5x groups performed slightly worse in training, indicating that subjects in those two groups took somewhat longer to locate and lock the bogey. With this variable, the greater the range at which the bogey is identified and locked, the better opportunity a pilot has to make decisions. In transfer, the 1.0x and 1.5x groups continued to improve, however, the mixed group showed a significant decrease in the first transfer trial block ($F_{3,20}$ = 37.64, $p <$ .001)(see Figure 2). This latter finding could be due to the relative uncertainty of the initial closure speeds and range-to-target caused by mixing the accelerated conditions.

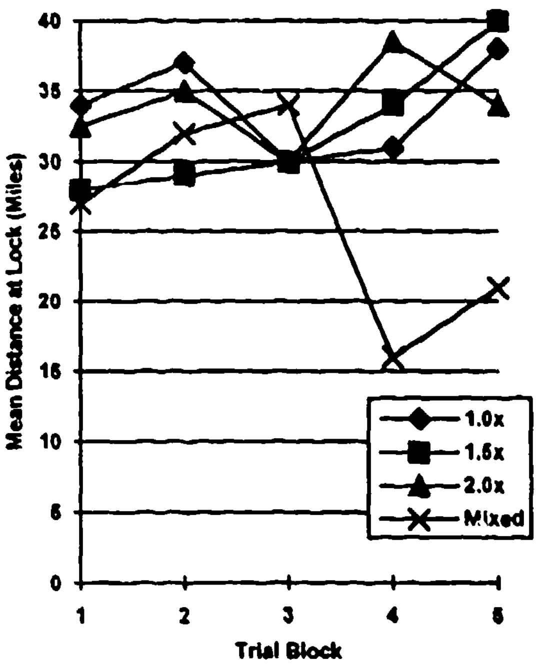


Figure 2. Mean distance at lock by trial block

For the stern conversion score, there are no significant differences between groups in training or between training and transfer performance among the four groups. The scoring procedure used for the stern task is based on a subjective rating that is often given by instructor pilots (IPs) to students. The score is based on assessing both the closure speed and aspect angle during the conversion. The idea being that when the pilot rolls-out behind the bogey (low aspect angle), the

pilot should not be more than three miles or less than one mile behind the bogey. As a rule-of-thumb, the closure speed should also be in proportion to the distance (e.g., at 2 miles, 200 knots closure speed). Although not statistically different, there is an actual decrease in performance from the last training block to the first transfer block followed by a slight increase in performance at the last transfer block. In the end, performance for the 1.0x group is higher than the other groups. The results of the stern conversion, taken together, tends to suggest that piloting tasks that involve well-learned (at real-time) and continuous responses to both internal (ownship) and external (bogey) positioning cues might not benefit from above-real-time simulation.

For the air combat maneuvering (ACM) task, time to first lock, time to reach criterion, and number of valid missile shots were analyzed by group. For time to first lock, which is a measure of the speed at which a pilot acquires his adversary on radar, all groups except the 1.0x group saw a significant increase in lock time from the last training block to the first transfer block ($F_{3,20}$ = 2.92, $p <$ .05). In comparing the groups at the final transfer block, both the mixed and 1.0x groups performed significantly better than either the 1.5x or 2.0x group. The 2.0x group also outperformed the 1.5x group in transfer (see Figure 3).

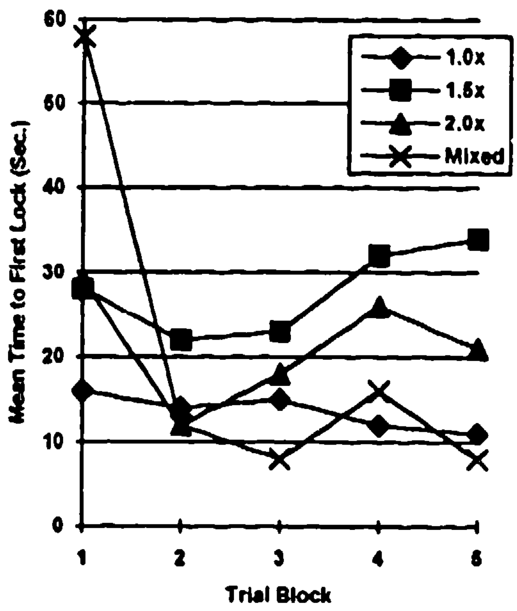


Figure 3. Mean time to first lock by trial block

For the time to reach criterion, there was no significant difference between groups from training to transfer. In comparing the last transfer block, however, the mixed group performed significantly better than either of the other groups ($F_{3,20}$ = 4.55, $p <$ .014) (See Figure 4).

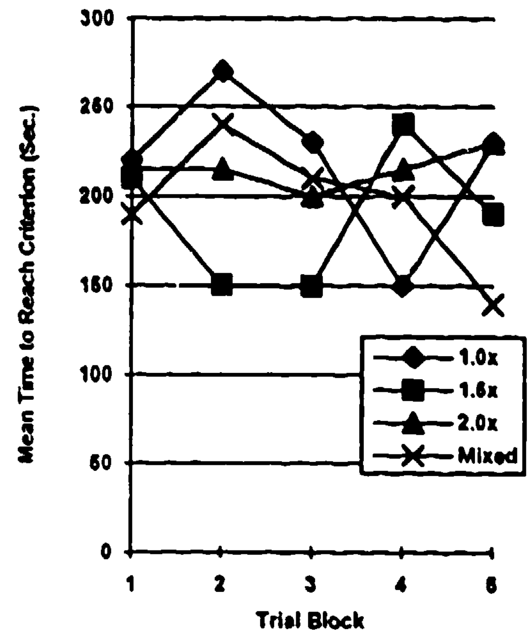


Figure 4. Mean time to reach criterion by trial block (ACM)

Finally, the mean hit/miss percentage were analyzed and revealed no significant differences between group in either training or transfer. Upon further inspection, it was apparent that this metric was somewhat biased due to the performance of the missiles. This point is expanded in the discussion section below.

DISCUSSION

The EP results demonstrated that all the groups trained under accelerated time conditions produced significantly higher accuracy in performing an emergency procedure in the transfer condition than did a real-time control group. The mixed and the 2.0x groups performed the EP near perfectly (100% and 96.6%, respectively). The 1.5x group's accuracy was almost 90%, while the control group scored the lowest at about 72%. This finding in particular demonstrates that ARTT may have potential to train procedural tasks with greater accuracy and in less time. In the EP task, the difficulty of the task was increased by placing all groups under the additional (simulated) stress of

having to perform the EP during a secondary air combat task An unexpected result was in each ARTT group, the number of enemy MiGs killed was six times higher than the 1.0x groups when compared in the real-time transfer blocks. There was also no significant difference between the groups when analyzing the time to complete the EP variable. The subjects, after a few trials, mastered the procedure and their performance stabilized. This seems to indicate that ARTT does not necessarily effect the speed with which pure motor tasks are performed.

Results of the stern conversion tasks are less clear, and neither support or refute the ARTT concept. For this task we attempted to implement ARTT by increasing the velocities of the ASAT and the bogey. In retrospect, due to the physics and geometry of the stern task, we failed to create a savings or reduction [illegible] time which is a central tenet in ARTT. The [illegible] forced the ARTT groups to take essentially the [illegible] in training as the real-time control group. [illegible] experiments we have been successful by speeding up targets, ownship, or both. This was not the case for the stern task. Moreover, pilots differ greatly in their approach to performing the task. Some would perform a low/high or high/low vertical conversion while some would initially offset left or right and perform a "standard" conversion. This made it difficult to establish useful measures of performance. Tasks such as the stern conversion that could be performed successfully using one or more alternate strategies, did not produce useful measures.

The air combat maneuvering (ACM) task also produced mixed results. Again, the fact that pilots have different flying styles leads to difficult performance assessment. The pilots were instructed to take two valid face shots - one at each bogey. A "valid" shot was one in which the range from the bogey was less than or equal to six miles and the aspect angle was between 135 and 180 degrees. The ASAT software modeled only the older AIM-9J and AIM-9L missiles. Unfortunately, when the raw data was inspected, it became clear that the pilots had great difficulty achieving "valid" missile shots, as they were defined, regardless of the group they were assigned. The explanation for this phenomenon lies in the performance of the missiles and the attack profiles preferred by the pilots. Specifically, the AIM-9L is capable of a high aspect kills, but its performance is significantly worse than the newer AIM-9M which the pilots are familiar with. The hit/miss percentage metric, therefore, cannot be considered a true reflection of pilot/weapon performance. In addition, most pilots chose to "offset" or break right or left to create more of an advantageous aspect angle. With a less than optimal high aspect kill performance of the AIM-9L missiles, the fight usually degenerated into a tail chase with a time savings disappearing since both the ASAT and the MiGs were both accelerated.

There were some trends in the ACM task that, although are not statistically significant, bear some mentioning. The mixed group were 11% faster in disposing of the two MiGs. The mixed group also showed the fastest reduction in time to first lock from training to transfer. Finally, the hit/miss percentage score was highest in the 1.5x and 2.0x groups.

CONCLUSION

Based on the results of this research, tasks that contain simple psychomotor or procedural components such as the emergency procedure task performed on the F-16 ASAT clearly benefit from ARTT. Moreover, this research demonstrated that task type and task content are differentially affected by ARTT. The ARTT groups showed higher performance scores when compared to a real-time control group in transfer for the EP task. For tasks with more complex cognitive components such as the ACM and stern conversion, there was no clear advantage in the ARTT groups compared to a real-time control group. The stern and ACM tasks allowed for alternative performance strategies that pose particular measurement and interpretation problems.

The increase accuracy of performing EPs bears further study because of the obvious implications for safety training. Many real-world emergencies require accurate performance of checklist procedures under sometimes extremely stressful circumstances. In this study those trained under an accelerated condition not only performed the primary EP more accurately, they also were able achieve a significantly greater number of MIG kills (6x) on a concurrent secondary task.

With respect to the initial research objectives:

1. ARTT was more effective than conventional real-time training in the case of EP task. The stern conversion and ACM task results were mixed.

2. For those significant effects, the group that provided the greatest performance improvements was the one that mixed the presentation at different speeds. This supports the contention that task variety in training leads to higher performance.

3. The impact of the ASAT study on training time is inconclusive due to methodological considerations.

Finally, as expected none of the ARTT groups experienced any negative transfer of training to real-time transfer tasks.

The results of this experiment can be seen as further support of the benefits of training at Above-Real Time. The emergency procedure task results illustrate the performance increases obtainable using ARTT on existing simulators. The other two tasks did not restrict the pilot's actions sufficiently to allow useful measures to be obtained. American pilots are arguably the finest pilots in the world, but their independence and cunning that make them great, also makes them difficult to restrict and measure. Consider the evolution of research listed below:

- The first use of "fast time" or ARTT in simulators was Jack Kolf at NASA Dryden over 20 years ago. (Kolf 1973).
- NASA'S initial success was followed by successes in the lifting body program as well. (Hoey 1976).
- The success of ATC by the FAA study. (Vidulich 1983)
- Success of VIGS time saving and performance increase. (Guckenberger 1992)
- Emergency procedure in F-16 accuracy increase

Applications of ARTT to simulators seems to have merit. The theoretical frame work for ARTT continues with synthesis from many diverse fields, most notably audio perception who's relativistic working models may transfer to illuminate ARTT's working relativistic model.

ARTT and the intrinsic time adaptability of man is a vast field of great potential.

FUTURE RESEARCH DIRECTIONS

Near-term work will focus on expanding the application of ARTT for emergency procedure training. We are also beginning to explore techniques to test the effectiveness of ARTT on subsequent performance in the actual aircraft.

The overall aim of the ARTT concept is to exploit the time adaptability of humans and foster a new way of thinking about time manipulation in the man-machine interface. Future research directions might include safety, education, medical, and entertainment applications. For example, it would be possible to increase the voice and data communication rate over a network to allow crews or teams to train at faster than real-time. Time flow could be controlled for the benefit of the trainee. New training methods that are *time flexible* will change form, fit and function of the man-machine interface. ARTT programs are initially planned in simulation and training with follow on to use ARTT in the man-machine interface. Emergency procedure training for pilots, both commercial and military is envisioned as the initial proving ground.

Current Research Projects include:

- ARTT for airborne weapons training
- Virtual Time Adding the fourth Dimension to Virtual Reality: Next Generation Man-Machine Interfaces
- Above Real Time Communication
- ARTT theoretical model: Relativistic Time-Speed Reading-Speed Listening -> Speed Simulating

REFERENCES

Cohen, J. (1964, November). Psychological time. *Scientific American* pp. 116-124.

Fraisse, P. (1984). Perception and estimation of time. In M.R. Rosenzweig & L.W. Porter (Eds). *Annual Review of Psychology, 35* 1-36.

Guckenberger, D., Uliano, K.C., & Lane, N.E (1992). *The application of above real-time training for simulators: Acquiring high performance skills.* Paper presented at the 14th Interservice/Industry Training Systems and Education Conference. San Antonio, TX. , 928-935

Hoey, R. G., (1976). *Time compression as a means for improving the value of training simulators.* Unpublished manuscript.

Hoffman, R.G., & Morrison, J.E. (1987). *Requirements for a device-based training and testing program for M1 gunnery: Volume 1: Rationale and summary of results* (Report No. FR-TRD-87-41). Alexandria, VA: Human Resources Research Organization.

Jones, Mari R. (1976) Time Our Lost Dimension *Psychology Review* Vol. 83 (5) 323-355

Hahn J. & Jones M. J. (1981) Invariants in Auditory Frequency Relations, Scandinavian Journal of Psychology 22, 129-144.

Kolf, J. (1973). *Documentation of a simulator study of an altered time base.* Unpublished manuscript.

Lane, N.E. (1987). *Skill acquisition rates and patterns: Issues and training implications.* New York: Springer-Verlag.

Lee, T.D., & Magill, R.A. (1983). The locus of contextual interference in motor skill acquisition. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 9,* 730-746.

Matin, E., & Boff, K.R. (1988). Information transfer rate with serial and simultaneous visual display formats. *Human Factors,* 30, 171-180.

Parasuraman, R. (1986). Vigilance, monitoring, and search. In K.R. Boff, L. Kaufman, & J.P. Thomas (Eds.), *Handbook of perception and human performance: Volume II: Cognitive processes and performance* (pp. 43-1 - 43-39). New York: Wiley.

Schneider, W. (1985). Training high performance skills: Fallacies and guidelines. *Human Factors, 25,* 285-300.

Shea, J.F., & Morgan, R.L. (1979). Contextual interference effects on the acquisition, retention and transfer of a motor skill. *Journal of Experimental Psychology: Human Learning and Memory, 5,* 179-187.

Shiffrin, R.M., & Schneider, W. (1977). Controlled and automatic human information processing: II. Perceptual learning, automatic attending, and a general theory. *Psychological Review, 84,* 127-190.

Skelly, June J. (1993) *The Role of Event Time in Attending* Time & Society Vol. 2(1), 107-128

SPSS, Inc. (1992). *SPSS for Windows* [Computer Program]. Chicago, IL: SPSS, Inc.

Stevens, S.S. (1975). *Psychophysics: Introduction to its perceptual, neural, and social prospects.* New York: Wiley.

Thompson, M. (1965). *General review of piloting problems encountered during simulation flights of the X-15.* Paper presented at the 9th Annual Meeting of the Society of Experimental Tests Pilots.

Toumodge, S.S. (1990) *Signal and Data Processing of Small Targets* Paper presented at the Orlando April 1990 Annual Meeting of the Society of Photo-Optical Instrumentation Engineers. (A91 - 36901 15-32) 378-385.

Vidulich, M., Yeh, Y.Y., & Schneider, W. (1983). *Time compressed components for air intercept control skills.* Proceedings of the 27th meeting of the Human Factors Society. 161-164.

ACKNOWLEDGMENTS

This research effort was made possible by a grant from the NASA Dryden Flight Research Center (DFRC). Special thanks to Vern Carter of Northrup for his kind assistance in tracing the history of "fast time" simulation back to Jack Kolf at NASA DFRC. Mr. Kolf was the contract monitor for this effort, and he was the first to originate the concept of "fast time" simulation while working on the NASA X-15 project. His encouragement and support of our research effort began long before the actual contract. His prior experience and suggestions were invaluable.

The authors would like to express their deep appreciation to the pilots of the 56th Tactical Training Squadron, MacDill Air Force Base, Tampa, Florida.

INSTRUCTIONAL DESIGN ISSUES IN DISTANCE LEARNING

William J. Walsh, Elizabeth G. Gibson, Patricia Y. Hsieh
Mei Technology Corporation
San Antonio, Texas

Dennis Gettman
Armstrong Laboratory
Brooks AFB, Texas

ABSTRACT

This paper reports preliminary results of research into distance learning currently being conducted by the authors. Although some of the results reported here are preliminary, the trends identified should not change significantly.

While many organizations conduct distance learning programs, there has not been much focus on issues of instructional design specifically directed towards distance learning. In this paper, current research on trends in instructional design issues pertaining to distance learning are investigated. Focus of the research was on evaluating the delivery of hands-on technical training via distance technologies. Data is presented on the impact of distance learning on the curriculum, types of student - instructor interaction, student interaction with the instructional materials, and on the preparation of faculty and staff for distance learning.

ABOUT THE AUTHORS

William J. Walsh has been involved in the design and development of training systems, and researching training technology issues for over 15 years. He has worked on and managed programs involving various implementations of training technology, including computer-based training, multimedia applications, intelligent computer-assisted training, simulations of maintenance and troubleshooting, and distance learning among others. Recently his concentration has been on technological applications to reduce instructional development time and increase instructional quality.

Elizabeth G. Gibson has over ten years of experience in research design, administration, education and training. She has been involved in large scale multidisciplinary projects integrating behavioral and physical sciences including psychological and organizational evaluation of applied behavioral science methods for assimilating, organizing and evaluation of complex social systems data. She has a Ph.D. in Biological Anthropology from the University of Oregon.

Patricia Y. Hsieh has nine years of experience in instructional design and classroom teaching. She has worked with a variety of training technologies, including computer-based training, embedded training, interactive video and intelligent tutoring systems. She has an M.A. in Instructional Technology from the University of Texas and a B.A. from Harvard University.

Dennis Gettman is a research psychologist at the United States Air Force, Armstrong Laboratory, Human Resources Directorate. He is the technical project manager for emerging training technologies, including computer-based training, automated job-aiding, performance support systems, etc. He recently coordinated and hosted an international workshop on distance learning and collaborative learning.

INSTRUCTIONAL DESIGN ISSUES IN DISTANCE LEARNING

William J. Walsh, Elizabeth G. Gibson, Patricia Y. Hsieh
Mei Technology Corporation
San Antonio, Texas

Dennis Gettman
Armstrong Laboratory
Brooks AFB, Texas

INTRODUCTION

Distance Learning is defined as: *any method of presenting training that is interactive and in which the students are physically separate from the instructor* (AETC, 1991). The training environment in the Air Force is changing. Efforts to decentralize training management and export training through distance learning and other technologies have already begun, and this trend will surely continue. Previous Armstrong Laboratory research in computer-based training revealed that the Air Force had not been thoroughly prepared for implementation of computer-based training (Walsh, Yee, Grozier, Gibson & Young, 1992). This paper will report the results of a study conducted by the authors for Armstrong Laboratory to determine issues to aid the Air Force in preparing for implementation of distance learning.

Past Research Focus

Although distance learning technologies have been utilized for many years,[1] distance learning has recently come under increased scrutiny as a viable technology for technical training. From a preliminary review of the literature it appears that the bulk of research has focused on distance learning technologies employed and how various organizations have implemented these technologies (presumably successfully). Few studies have concentrated on student-related issues or instructional quality concerns.

The current study builds upon previous research work in distance learning conducted by Air Education and Training Command (AETC) to assess the state-of-the-art in distance learning technology (AETC, 1991). Using the AETC study as a starting point, the authors surveyed organizations involved in distance learning to determine how far the technology has advanced, and to assess the organizations' experience with distance learning. Of particular interest were lessons learned concerning the impact of distance learning on the quality of the curriculum, student - instructor interaction, student interaction with instructional materials, and on unique approaches to developing instructional materials for distance learning technologies.

Selected organizations were contacted to determine how their approach to distance learning affects the preparation and training of their instructional staff, whether they employed any special techniques to select, prepare or modify old instructional materials or create new instructional materials for implementation in distance learning, to assess the effectiveness of their programs, and to gauge the organizational impact of distance learning.

Research Goals

A majority of training which takes place in the Air Force is directly related to maintenance functions. Even training which is not maintenance related is primarily task (skill) oriented. Nearly all of this training requires some kind of hands-on experience with aircraft, weapon systems or equipment. For any technology to have a significant effect on Air Force technical training it must be able to adequately address the requirements of these hands-on components. In our opinion, effective technical training requires sound instructional design strategies as its basis. Therefore, we have sought to identify distance learning programs which have addressed similar training requirements effectively.

A primary goal of this research was to determine if there are specific categories of

[1] 38.1% of respondents to our survey indicated they have had programs for 10 years or longer.

objectives, task characteristics, or instructional strategies which lend themselves to particular distance learning technologies. We have attempted to answer Miller's (1990) question: What can be done better through distance education than in a classroom? Some emphasis is placed on the use of mediated instruction, i.e., distance learning using computer-assisted instruction, to assess its applicability and effectiveness. While much distance learning tends to consist of video teletraining with an instructor presenting the learning materials to students at remote site(s) over television transmissions, the research team questioned if this method was the primary strategy for distance learning, and whether it was the best one for hands-on objectives.

APPROACH

The research approach taken was to determine the state-of-the-art in distance learning from the literature, to glean from the literature specific problems which were of interest to the Laboratory, and to design a survey of distance learning organizations to assess their approach to these issues.

Literature Review

Prior to designing a questionnaire, an extensive literature review was conducted. The purpose of the literature review was to identify current trends in distance learning, to determine what potential research issues might be, and to assess if there had been any previous research conducted which might offer solutions to the instructional design issues. Some of the literature reported success stories for individual distance learning programs (Griffin & Hodgins, 1991, Chung, 1991, Heathman & Kleiner, 1991, McKell, Hardy & Stocks, 1992, and Marshall, 1991, among others). While reviewing this large volume of literature, we found that there were unresolved problems associated with instructional effectiveness of distance learning for certain kinds of skills. This provided several topics for questionnaire development.

The literature was also a source of data regarding approaches, methods and techniques which might offer success if applied to distance learning. We examined these carefully whenever we visited one of the organizations later in the study. Practices which could be of benefit to the Air Force in designing instructional strategies for distance learning were of primary interest to the research team during site visits.

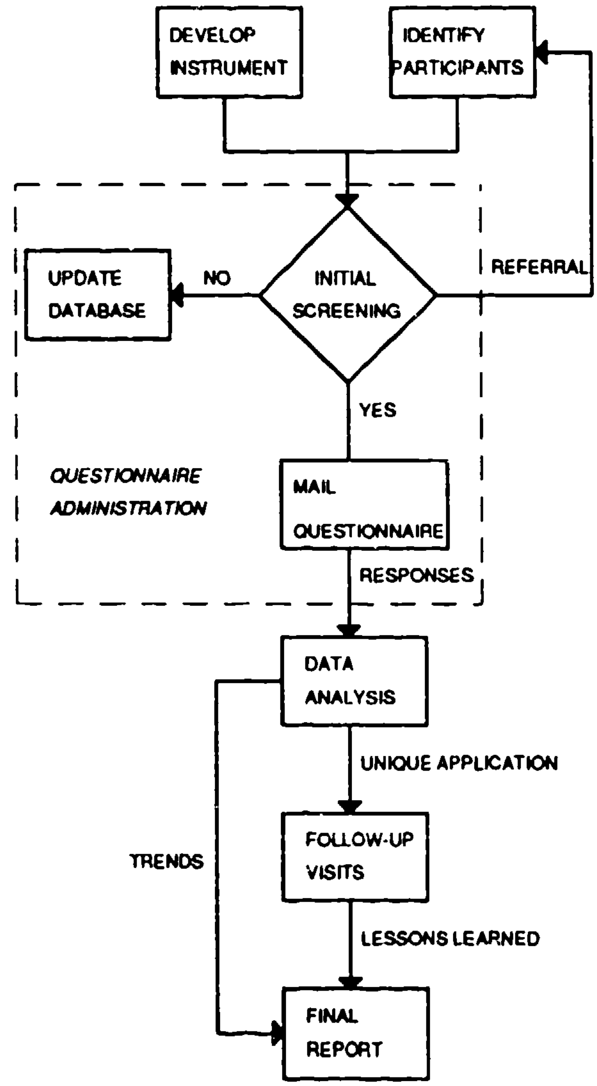


Figure 1 GENERAL APPROACH

Survey Development

After identifying potential distance learning issues and problems from the literature review, the team developed a questionnaire to be used in surveying the field. The questionnaire consisted of 65 questions organized into 5 major areas: organizational profile, student information, faculty information, instructional design and general information. Within each of these areas several topics were examined. The questionnaire was designed to provide the respondents with a

number of choices in each category, yet it did not limit them from providing specific comments to a question if the categories listed did not represent what they were doing.

Identification of Distance Learning Organizations

Since our definition of distance learning was broadly inclusive, there was no single source which provided us with a list of all or most distance learning organizations to be surveyed. We were interested in distance learning providers, but not merely instances of a school district implementing televised classes in math, science, etc., or in technology vendors interested in selling their systems. Rather, we wanted to contact as many organizations which had tailored their curriculum, did something specific to prepare their instructors, or had some unique aspect to their distance learning program. We were especially interested in organizations which might be providing hands-on technical training. The research team developed a database of distance learning organizations from various distance learning related sources such as: catalogs, networks, journal articles, the AETC study, and (probably most effective) referrals from other organizations. While we would not assert that this listing is comprehensive, it is a broad sample of organizations which are providing distance learning services.

Conduct Survey

The survey was conducted over a 3 month period. During this time all of the organizations in the database were contacted by phone. Each organization which had programs of potential interest was sent the questionnaire for completion. If respondents did not return the questionnaire within the allotted time, they were called again regarding the status of the questionnaire. Slightly more than 70% of the questionnaires were returned.[2]

Initial Screening -- Our approach included an initial telephone screening of potential respondents. During the screening we asked several organizational profile questions which were indicators of the kinds of programs being conducted. Based on participants responses to these questions, they were sent a copy of the questionnaire to complete. Frequently, calls were made in the blind, i.e., we had no definite contact at the organization, and the research team was forced to track down the right person to respond to the questionnaire. Very few (2.7%) of the organizations called (n=187) refused to participate in the survey by even answering the screening questions.

Questionnaire Distribution -- The questionnaire was distributed to the participants over a three month period. Participants were given a date by which to return the completed questionnaire. If a questionnaire was not returned within 1 week of the deadline, a follow-up call was made to the point of contact. In most cases the participants indicated at that time whether or not they would return the questionnaire.

Data Analysis -- The data analysis focused on providing descriptive data and comparing relationships at the aggregate respondent level. For issues of special interest, data were collapsed and assessed for specific demographic groups. Numerous qualitative data were gathered from two open-ended questions. Whenever possible these data were reduced into a more manageable form by categorizing responses according to the same broad areas as the questionnaire.

Expected Outcomes -- We expected the data analysis to provide us with information concerning: trends in distance learning, indications of unique approaches to curricular materials, potential novel approaches to instructor - student interaction, successful use of distance learning technology for hands-on training, and other similar items which could contribute to preparing the Air Force for implementation of the technology. We also hoped to determine if relationships existed between the types of skills, tasks or objectives trained and the appropriateness of various distance learning media.

Follow-up Visits -- The data analysis was also designed to provide indications of candidate sites for follow-up visits. These sites were to be determined based on their unique or successful application of distance learning technology to

---

[2] Questionnaires were sent to 182 organizations, of these 128 were returned.

training. The research team identified the more promising organizations, ranked them according to several factors (uniqueness of the program, potential for lessons learned, willingness to host a visit, proximity to each other, etc.), and recommended the list to the Laboratory. Upon approval of the list by the Laboratory sponsor, visits were scheduled to each of the organizations.

Prior to visiting the selected organizations, the research team developed a set of visit protocols. These protocols consisted of verification of the data provided in the questionnaire, details of the specific research questions to be answered, and complete information to be gathered about the program or approach. With this in hand members of the team accompanied by a Laboratory sponsor visited the selected organizations and conducted the follow-up interviews. Most organizations were eager to demonstrate their programs to the team. These visits provided extremely useful anecdotal information about distance learning which, when coupled with the results of the questionnaire, provide some insight into distance learning implementation.

Document Research Issues

While much interesting and useful information about distance learning was acquired during this research, its purpose was not to develop definitive guidance for Air Force implementation of the technology. Rather the outcome of the research was to identify current trends and problems which will need to be overcome so that future implementations of distance learning technology can be successful. As a consequence, during the study we documented: 1) instances of successful (or partially successful) applications of distance learning, 2) methodological deviations from traditional instructional design and development techniques which may be necessary to take full advantage of the technology, 3) problems encountered in applying distance learning to specific types of skills, tasks or objectives, and 4) specific research issues which should be explored further.

Comments by Industry, Academia & Government -- As a final step in the research, the findings were scheduled to be presented to a panel of international distance learning practitioners and researchers. Comments by this group will be used to refine the Laboratory's distance learning research agenda.

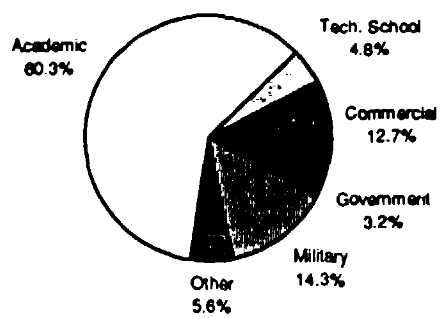
FINDINGS

Our initial findings cluster around the five major areas of the questionnaire, namely, type of organization, student, faculty, instructional design and problems encountered. We will discuss each of these areas below.

Characteristics of Organizations Involved in Distance Learning

We surveyed a broad range of distance learning providers from academic institutions to commercial firms providing employee training via distance learning. The various types of organizations responding to our survey are indicated in Figure 2.

Figure 2. Organizations Conducting Distance Learning



The distance learning providers we surveyed did not restricted their programs to a single medium. Nearly all (99.2%) report using several media in their distance learning programs with 73.8% reporting using five or more. While computer-based training is popular (58.7%), video broadcast, either live or taped, is even more frequently used (73.8%). 84.6% of respondents indicated that special equipment is necessary for students taking distance learning courses. Of those courses requiring special equipment, 48.8% of respondents said that they provide it to the students.

Organizations involved in distance learning tend to have large programs with 51.2% reporting

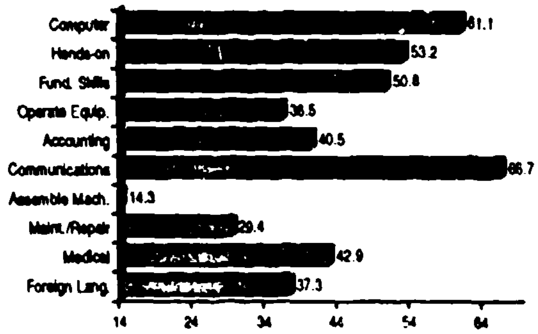
21 or more courses offered, and 88.9% with more than 100 students. Courses range from very small, 1-10 students (20%), to very large, over 100 students (15%). While many organizations reported that there were course enrollment limitations, 38.9% said that there weren't any limitations on enrollment. Many organizations (63.7%) indicated that their distance learning courses were substantially the same as the conventional courses they offered. In spite of this, only 7.1% indicated there was no need to modify the approach for distance learning.

The tendency to use multiple media appears to be reflected in the wide variety of skills trained. Distance learning programs appear to be used for everything from teaching hands-on skills such as assembling machinery (14.3%) to communications (66.7%). Figure 3 displays some of the broad spectrum of skills taught via distance learning technology.[3] The research team was surprised at the high numbers for hands-on skills (53.2%), equipment related training (36.5%) and maintenance and repair training (29.4%). These indicated to us that distance learning technology has at least some capability to deliver the same kind of training as Air Force technical schools. While some of the distance learning programs were conducted in a conventional classroom setting (44.4%), many take place on-the-job or at the worksite (55.6%). While the largest percentage of respondents indicated they trained problem solving skills (73.8%), in fact, the same percentage indicated that they trained job related skills.

In general, a thumbnail sketch of the typical distance learning provider is one with several courses taught by conventional means as well as distance technology. In addition, this typical organization also offers several unique courses via distance learning. No restrictions are placed on the type of skills taught by distance learning. In fact, it appears that hands-on skills are as likely a candidate for distance learning as cognitive skills.

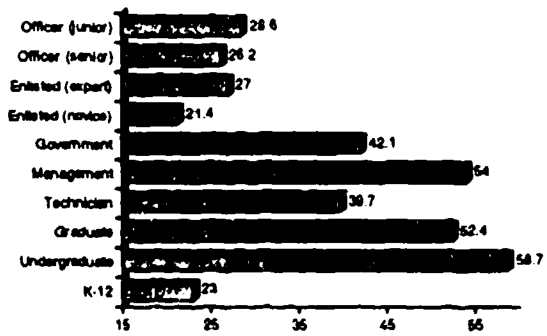
Characteristics of Distance Learning Students -- Distance learning programs are

Figure 3. Type of Skills Trained



available to a wide variety of students (see Figure 4). In general, students have the option of selecting either distance learning or a conventional course (69%). Our respondents indicated that the primary reason students had for selecting distance learning was that it was more convenient (77.5%).

Figure 4. Distance Learning Students



One issue of importance to the Laboratory was the amount and kind of interaction between students and instructors in distance learning. Although many programs (62.4%) indicated that this interaction took the form of written correspondence, e.g., homework, tests, exercises, etc., several other means were also used such as questions and answers from the distant classroom (60%), telephone calls during conference hours (58.4%) and computer link, e.g., via modem (43.2%). It is interesting to note that 82.2% of the respondents indicated that students had at least two or more ways to interact with instructors. While interaction intervals may vary from course to course, 42% of the

[3] In this example and throughout this paper some percentages will add up to more than 100%. This is due to the fact that a respondent could check several options for some questions.

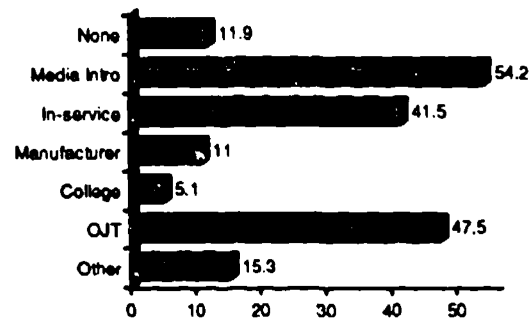
respondents indicated that it was frequent, i.e., daily or weekly.

Another point of interest for the research team was whether distance learning offered some efficiency in achieving learning goals over conventional instruction. Our d..a did not indicate that such efficiency was being achieved on a large scale with 8.9% of respondents reporting that distance learning courses are longer and 12.9% reporting that distance learning courses are shorter than conventional courses. The majority of respondents (55.6%) reported that the distance learning course is a fixed length; 47.1% said it is the same length as the conventional course. When we asked if students took longer to complete the distance learning course, many respondents indicated that they take the same time in either course (47.1%).

We were also interested in finding out if novel approaches to performance assessment were being used in distance learning courses. Many respondents (42.7%) indicated that course materials included self-assessments. However, 74.2% of respondents indicated that written tests were used for student assessment. Only 16.1% indicated that performance tests were administered on-the-job or elsewhere (12.9%). In general, other than written tests, performance assessment is substantially the same as conventional classroom with 56.5% using periodic work assignments, 54.8% relying on instructors monitoring student work and 34.7% relying on verbal examination of the knowledge.

Characteristics of Distance Learning Instructors -- Most respondents reported the distance learning faculty is the same as for conventional courses (72.6%). Only a few respondents (12.9%) indicated that they used a special group of instructors from their own staff for distance learning courses or specialists from outside the organization (21.8%). Most of the instructors were chosen because of their teaching experience (60.3%) or their experience with the media (36.4%), although 50% indicated that the faculty had no special background, rather they were selected from those available. Several respondents indicated the kind of training which distance learning instructors receive (see Figure 5.).

Figure 5. Instructor Training



We also asked what was included in the instructor training programs. As expected most said that they provided an introduction to distance learning technology (68.9%). Equally important were how to make use of media in a distance learning environment (63%), communications skills (63%), and how to deliver the subject matter via distance learning (59.7%). What we thought might be two of the more critical skills also received treatment: how to operate the distance learning equipment (52.1%), and how to promote distance learning interaction (56.3%). Apparently it was less important to provide training in how to evaluate distance learning students (33.6%). Many programs also provided training in instructional development skills for distance learning (49.8%).

While various reasons were offered as the reason for training distance learning instructors, the one cited most often was that untrained instructors were not effective (44.9%). The training programs appear to be effective since respondents reported that instructors made better use of media (53.8%) and interaction strategies (52.1%). Most respondents (59.8%) felt that instructors were better able to utilize features of distance learning technology after training.

Characteristics of Distance Learning Curriculum -- Many programs appear to rely on conventional courses as the source for their distance learning curriculum. In fact, the distance learning curriculum is frequently the same as conventional courses (58.1%), or the conventional curriculum is specially adapted for distance learning (49.2%). Less frequently (36.3%) is a

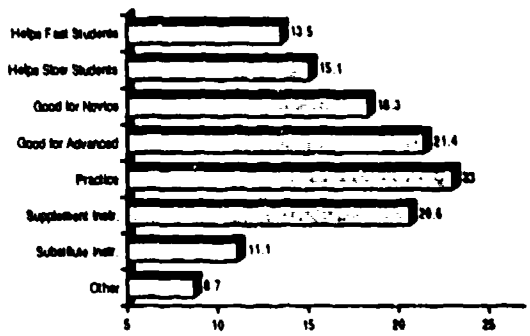
course developed specifically for distance learning. While distance learning courses may be derived from conventional courses they are far from stable; only 12.9% of respondents report that the distance learning curriculum is very stable. Rather, the distance learning curriculum evolves every time it is taught (30.6%). Perhaps this may be due to the fact that it is based on a conventional course which does not take full advantage of distance learning capabilities.

As pointed out earlier, one of the Laboratory's research goals was to determine if specific types of tasks or objectives fit distance learning better than others. Few respondents (11.4%) said that there was a certain category of objectives which was best for distance learning. Some (13.8%) indicated that they developed objectives especially for distance learning. In general, objectives were no different than conventional course objectives (69.1%), or were simply conventional course objectives adapted for distance learning technology (33.3%). Nor did organizations appear to have an accepted methodology for selecting distance learning media for objectives. When asked why they selected distance learning for certain objectives respondents tended to use objectives from existing courses (70.3%). Only some indicated that they performed some kind of media analysis (16.1%), based the selection on research (11%), or had a previous model (11%).

Computer-Based Training in Distance Learning -- Many respondents (56.6%) indicated that they used computer-based training as part of their distance learning curriculum. However, 39% of respondents found computer-based training to be an effective distance learning tool. Various reasons were provided as to why it is effective (see Figure 6). Only 17.2% of respondents used it as the primary method for delivering of instruction.

Curriculum Development -- Curriculum development for distance learning is not easily classified. Organizations most frequently report using approaches such as instructional systems development (ISD) (33%), or their own development process (35.1%). According to the respondents they use a specific curriculum development process because there is a preference for it among the faculty/instructional

Figure 6. Why CBT is Effective Tool



development staff (49.5%), or it takes distance learning requirements and capabilities into account (57%).

When a special curriculum is developed for distance learning it has definite characteristics such as including additional graphic media (70.4%), and increased opportunities for student interaction (64.3%). Frequently some materials are provided to students in advance (62.2%). There is generally, more independent student work (43.9%) and more frequent assessment of student performance (40.8%).

In an overwhelming number of cases (73%) the course instructor develops the distance learning curriculum. Far fewer organizations rely on a staff instructional designer specially trained in distance learning (38.5%), and even fewer rely on outside organizations (13.1%) or consultants and contractors (14.8%). In general, the majority of respondents (52.5%) reported staff with more than 3 years experience in distance learning. However, some (18%) indicated staff members with no experience or less than 6 months. Just as for distance learning instructors, most organizations also provided training for curriculum developers (68.8%). While a few seemed to pick up training from other sources, college courses account for the bulk of the outside training (46.4%). Again, just as for the instructors, the training was effective. 43% of respondents reported developers made better use of media. Developers were better able to design interaction strategies (48.6%), and utilize features of distance learning technology (54.2%). 76.8% of respondents reported using a combination of professionals in developing the distance learning

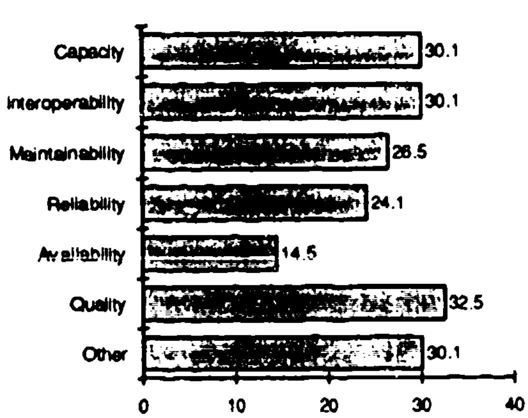
curriculum including instructor (84%), instructional developer (63%), media specialist (56%), distance learning technician (48%), or some other (23%).

We asked the respondents how long it took them to develop distance learning courses. Generally, they spent varying amounts of time performing the various activities associated with curriculum development. Although some (43.6%) indicated that it took as long as conventional courses, many (55.4%) said that media preparation took longer. 14.3% of respondents have formulas they use in developing distance learning courses and provided them to us.

Problems Associated with Distance Learning

Surprisingly, 73% of respondents said they were conducting research in distance learning. The areas investigated range from the effectiveness of distance learning (58.2%), use of various distance learning technologies (54.9%), distance learning multimedia applications (41%), role of the instructor (40.2%), and curriculum development for distance learning (36.1%). Their research was reflected in the answers to questions we posed about distance learning problems. Nearly all (91.8%) reported some kind of problem. The principal ones were the availability of trained personnel (50%), and reluctance of the faculty to use the technology (54.9%). While there was some resistance to distance learning on the part of students (27%) and administration (32%), the real resistance came from the faculty (47.5%). The respondents were also able to categorize their technology related problems (see Figure 7).

Figure 7. Technology Related Problems



The reason most cited for the problems which the organization was having was inadequate funding (41.2%). Several other potential causes were reported such as lack of experienced or trained personnel (31.9%), limited or inadequate facilities (27.7%), inadequate time to plan and prepare (24.4%), and technological problems (25.2%). As one might expect, in the opinion of the respondents the solution which could have prevented or diminished the problem was additional funding (46.2%). Although several other solutions were suggested such as training for personnel (41%), management support (36.8%), public relations to overcome biases (35.9%), better facilities (26.5%), and additional time (28.2%).

Future Plans for Distance Learning

In spite of the fact that there are problems associated with distance learning, a majority of respondents (76.2%) indicated that they had plans to increase the number of courses. Other responses indicated similar positive attitudes toward distance learning. 72.1% of respondents plan to increase the scope of their distance learning programs and add or improve distance learning equipment. Many organizations plan to increase funding for distance learning (55.7%), train faculty in distance learning technology (44.3%), and improve distance learning facilities (54.9%). Only 1.6% said that they have plans to reduce or eliminate distance learning courses. Only 31.9% said they planned to add faculty. This corresponds with some responses we got which indicated that distance learning was cost effective because it could reach more students with fewer faculty. However, only 27.3% reported that the reason they are planning to make more use of distance learning was that it was cheaper than conventional courses. Respondents seemed to be impressed with distance learning course effectiveness (53.7%), and that students like distance learning (38.8%).

CONCLUSIONS

Obviously, from the problems reported training is needed for instructors in the use of distance learning technology. In particular how to provide for student interaction, methods of assessing student performance at a distance, and general communications skills. Currently, successful instructors tend to do more and less

successful ones drop out of the program. Curriculum developers also need to be prepared for distance learning. They must learn new ways of increasing student interaction with training materials, and how to make effective use of graphic materials to support distance learning. Some organizations have begun to develop guidelines for curricular materials based on their experience implementing programs. However, for the majority it appears that curriculum development for distance learning is a highly personal thing with each developer using and refining techniques which have worked in the past.

Costs do not appear to be out of line with other technologies or conventional instruction. While most organizations indicated that funding was a problem, it was not for the development of materials as is the case with computer-based training. Rather, distance learning costs are associated with equipment, satellite time, etc. In fact, the organizations we contacted indicated that distance learning was potentially a cost saver because it allowed more students to be taught by fewer staff, and delivered the instruction where the student was rather than on a central campus.

When we began this study we had the notion that we would find organizations that used a single distance learning technology exclusively or much more than others. In general, that is not the case. Distance learning programs tend to be eclectic in their approach to technology. In other words, they tend to use several complementary technologies together in their programs. This puts more emphasis on having a trained and experienced staff of curriculum developers and instructors so that they can take full advantage of the capabilities of various technologies.

Finally, there does not appear to be any systematic media selection process in use to identify what distance learning can do better than other forms of instruction. Still further, distance learning appears to be selected based on factors other than the kinds of skills to be trained. The ability to reach many students in dispersed locations is one factor which tends to be considered in selecting distance learning for a curriculum rather than characteristics of the training objective, the domain of knowledge to be learned, or the type of tasks to be performed. Further research needs to be done to determine just what distance learning is better for than other forms of instructional delivery.

REFERENCES

Air Education and Training Command, Training Technology Application Program. (1991). *Distance Learning: Final Report* (Contract No. F41689-89-D-0252). Randolph AFB, TX.

Chung, J. (1991). Televised Teaching Effectiveness: Two Case Studies. *Educational Technology*, 31 (1), 41-47.

Griffin, G.R. and Hodgins, M.M. (1991). VTT in the Navy: Training Now and for the Future. *T.H.E. Journal*, 12 (1), 65-67.

Heathman, D.J. and Kleiner, B.H. (1991). Training + Technology: The Future is Now. *Training and Development*, 45 (9), 49-52.

Marshall, B. (1991). A Training Success Story. *Training and Development*, 45 (9), 63-65.

McKell, L.J., Hardy, J.W., and Stocks, K.D. (1992). The IBM CENET System: Continuing Education for Management. *T.H.E. Journal*, 13 (4), 93-97.

Miller, G.E. 1990. Distance Education and the Curriculum: Dredging a New Mainstream. In *Contemporary Issues in American Distance Education*, ed. M. G. Moore. New York: Pergamon Press.

Walsh, W. J., Yee, P. J., Grozier, S. A., Gibson, E. G. and Young, S. A. *A Survey of Air Force Computer-Based Training (CBT) Planning, Selection, and Implementation Issues*, AL-TP-1991-0059, March 1992.

Acknowledgments - This research task was funded under Contract F33615-91-D-00651 by the U. S. Air Force Armstrong Laboratory, Human Resources Directorate, Brooks AFB, Texas. The authors also wish to thank Phoebe Farrell who conducted most of the telephone interviewing, and cataloging of responses. Kimberly Hicks and Todd Miller conducted statistical analyses of the results and offered useful advice as to how the results could be interpreted.

A DISTANCE LEARNING NETWORK CONTROL SYSTEM

C.N. Mckinney, Q. Tran, V. Gallegos and P. Gercken
IBM Federal Systems Company
9500 Godwin Drive Manassas, Virginia 22110

ABSTRACT

In the "Extending Classrooms to the Military Workplace" paper submitted last year, we discussed the benefits of distance learning over conventional training programs. We focused on hardware and introduced multimedia and a modular, building block architecture that supports distance learning and Computer Based Training (CBT) on one platform.

This year, we focus on implementation of a new system and we detail the software architecture. We performed a comparative analysis of several distance learning systems currently in operation and designed a new system incorporating the best features discovered during this investigation. This analysis identified a need for distance learning systems to use existing Department of Defense multimedia and networking technologies; provide capabilities for transmitting lessons to individual workstations; and provide features that allow one person to control the entire distance learning network.

This paper describes a generic framework for a distance learning network control system that allows one instructor to control the entire network operation and allows communication to receive sites over satellite, terrestrial, and Local Area Network (LAN) interfaces. The proposed control system supports two-way video, audio, and data transmissions between the broadcast and receive locations, and provides system monitoring capabilities from one central console. We discuss interoperability, open systems, and the functional requirements needed to control a distance learning classroom session. We describe basic software components of the distance learning control system: user interface, LAN control system, satellite control system, terrestrial control system, and receive site control system. We also define interface specifications and performance requirements, and define the relationship between components and subcomponents within the distance learning architecture. Finally, we give examples of how the implementation of the proposed control system can lead to reduced costs in developing, maintaining, and enhancing distance learning systems.

BIOGRAPHIES

Clifton McKinney is an Advisory Systems Engineer with the IBM Federal Systems Company (FSC), Manassas, Virginia. He now works in the Instructional Technologies New Business organization as Chief Systems Engineer. He writes proposals, develops architectures, and delivers distance learning and networked CBT solutions.

Quang M. Tran is an Advisory Software Engineer with IBM FSC in Manassas, Virginia. He was Lead Software Engineer for several training system projects at IBM FSC. Currently, he is a Software Engineer on the Close Combat Tactical Trainer contract.

Venisa Gallegos is an Associate Systems Engineer with IBM FSC in Manassas, Virginia. Her experience covers positions as a Programmer and Design Engineer on several classroom-based multimedia systems, including distance learning and networked multimedia classroom. She is now working on an independent research and development (IRAD) project to investigate video teleconferencing over a LAN.

Pam Gercken is an Associate Software Engineer with IBM FSC in Manassas, Virginia. She has been involved in several classroom-based training programs, including the interactive multimedia classroom, networked CBT systems, and distance learning systems.

A DISTANCE LEARNING NETWORK CONTROL SYSTEM

C.N. Mckinnay, Q. Tran, V. Gallegos and P. Gercken
IBM Federal Systems Company
9500 Godwin Drive Manassas, Virginia 22110

INTRODUCTION

Reductions in military budgets have forced the Department of Defense (DOD) and other federal agencies to develop cost effective methods of training and recertifying employees. The existing methods for training large groups of individuals is under attack due to instructor/student travelling costs and facility overhead expenses. In many cases the government has elected to train large groups of individuals using Distance Learning Systems (DLS). This paper describes an architecture that will allow the integration of DLSs into existing Local Area Network (LAN) based computer network environments. We discuss results obtained from surveys of existing DLSs and identify the primary features that should be provided by any DLS. We also explain why many of the standards defined to support an Open System computing environment are also applicable to DLSs. Finally, we discuss the cost benefits of implementing a DLS using our proposed architecture.

This paper focuses on the design requirement for a Distance Learning (DL) control system. We will not discuss the video and audio requirements of a DLS in detail. The selection of video and audio compression and transmission equipment is dependent on the user's picture and audio quality requirements. The proposed architecture allows users to select the appropriate video and audio equipment that meets their specific requirements without impacting the software control system.

DISTANCE LEARNING OVERVIEW

Distance learning is defined as the delivery of curriculum to students beyond the four walls of a conventional classroom. Unlike video teleconferencing, which requires as a minimum two-way video and audio transmissions, distance learning normally requires one way high quality video and two way audio. Since the primary objective of distance learning is to extend the learning environment to students at remote locations, it is desirable to provide methods for instructors to test the comprehension level of students. This testing feature can be implemented as an ad-hoc question initiation and results display capability or as a full feature test capability with student results being collected and stored for future analysis purposes. In either case, the system allows instructors to measure the effectiveness of the curriculum and student comprehension. To support this requirement, DLSs must provide a method for students and instructors to interact and allow instructors to list total student responses at receive sites (ad-hoc questions) or identify individual student responses (testing). Another DL requirement is to provide methods for students to "raise their hands" and ask a question. Just like in a conventional classroom, the instructor determines when a student can talk during the lecture. Therefore, the instructor should have the capability of answering questions or disregarding/canceling help requests. Finally, the DLS should provide a method for controlling transmission equipment and alerting instructors and/or administrators when errors are detected in the transmission equipment.

The ideal DLS is a system that allows an instructor to utilize the methods and techniques currently used to teach students in the conventional classroom environment. Hence, the training effectiveness and student comprehension levels being obtained in the current environment could also be obtained in a distance learning environment. The only differences in the two environments would be the location of the students.

Comparative Analysis

Over the last three years we have been evaluating IBM and other vendor developed DLSs. Most of these DLSs were a subset of a video teleconferencing system. That is, they provided one way audio and video transmission (instructor). There was no method for students to interact with the instructor other than a voice grade telephone line. Some systems provided Student Response Units (SRU) to allow students to answer questions; these devices also had microphone systems so students could request

help and talk to the instructors. Other systems offered two-way video for instructor and student return video transmissions. This system was usually a video teleconferencing system with a Multipoint Control Unit (MCU) which was used as a distance learning system. In most cases, these systems did not provide methods for instructors to ask questions and collect results. Although the technologies are available to support highly interactive distance learning systems, the design and implementation of these systems resulted in the following problems: too many people were needed to operate the DLSs; the DLSs were developed/integrated as stand-alone solutions; lack of modularity and growth capability; and limited transmission capabilities.

Many of the DLSs evaluated required at least three people to operate the system: instructor, broadcast site administrator and receive site administrator. In some cases another person was utilized to operate the multimedia equipment and monitor incoming calls from students. Our design will allow the entire distance learning network to be controlled by one instructor.

Advancements in LAN technologies and reductions in Personal Computer (PC) costs have led to a dramatic increase in the migration to LAN based computer systems. In addition, the migration from Host based systems to client/server environments has been a major contributor to the increased demand for PC networks. The potential result of this new environment is that every user will have a workstation as opposed to a terminal. Many of today's workstations can support multimedia upgrades to display full motion video and receive audio by inserting additional multimedia cards. Furthermore, since these workstations have processing power, software emulated SRU devices can be developed. The conclusion is that DLSs must be capable of operating in these new LAN based environments by utilizing existing multimedia machines as opposed to being a separate solution. This new DLS system is now a subset of an existing LAN based computer/training system, which can also support simulation, CBT, procurement, logistics and other military applications.

Many of the DLSs evaluated did not provide easy methods for upgrading their product. In cases where upgrades were possible, the upgrade had to be provided by the vendor because the hardware and software were proprietary. DLS features must be provided as modular upgrades to the baseline platform. These upgrades are not proprietary, but, whenever possible, should comply with existing and future hardware and software interface standards. In effect, the DLS should be developed on an Open System platform. Since the DLS is now a subset of the existing computer system network, it should comply with Open System supported standards such as: Portable Operating System Interface for Computing Environment (POSIX), Structured Query Language (SQL), Government Open System Interconnect Profile (GOSIP); Transmit Control Program Internet Protocol (TCP/IP); Distributed Computing Environment (DCE); Computer-aided Acquisition and Logistic Support (CALS); and H.261/Px64 (video compression algorithm).

Most of the DLSs evaluated only provided video and audio transmission to receive sites over satellite interfaces. If interactivity was supported, it would be implemented over a terrestrial interface using 56kbps, Packet Switching Data networks (PSDN) and/or voice grade telephone lines. None of the systems evaluated provided transmissions to receive sites over LAN interfaces. With new Wide Area Network (WAN) technologies and services becoming available, such as Asynchronous Transfer Mode (ATM), Switched Multimegabit Data Service (SMDS), Broadband Integrated Service Data Network (BISDN), and Synchronous Optical Network (SONET), it will be possible in the near future to transmit compressed video and audio economically over terrestrial lines. The new DLSs must be independent of the transmission medium; thus, allowing user to take advantage of existing infrastructure networks and expand when new technologies become mature.

DLS Functional Specification

Extensive trade studies and surveys of the DLS users community were conducted to compile a list of functions that should be provided in a generic DLS. The DLS functions are categorized according to the three types of DLS participants (operators): student, instructor, and administrator. For each type of participant, there is an associated workstation that provides the required functions. Detailed task analysis

were performed to ensure that the tasks to be performed by the student, instructor, and administrator were provided by the DLS.

Instructor

The instructor shall be provided with a multimedia station to deliver multimedia supported lectures and a workstation to control the DLS. The workstation shall be independent of the multimedia station; that is, the DLS shall operate independent of any multimedia authoring and delivery package. Furthermore, the DLS shall not be dependent on specific video/audio compression equipment. The instructor shall have the capability of utilizing laser discs, video cameras, text, computer graphics, digitized audio, VHS tapes, and animation to augment conventional stand-up lecture presentations. The DLS shall allow instructors to transmit high quality video, audio and network commands over satellite, terrestrial and/or LAN interfaces. Lecture material shall appear in a window on PC based student stations. Receive site return audio, video, and data shall be supported over terrestrial, satellite and/or LAN interfaces. The DLS shall provide six basic functions from the instructor workstation: system initialization, roster collection, question initiation, test initiation, student help processing, and system monitoring. Touch, mouse and keyboard entry capabilities shall be accepted by the instructor workstation.

The DLS shall provide a method of allowing instructors to assign receive sites to specific DL sessions. Configuration files shall be created in ASCII formats and shall be protected using passwords and encryption. These files shall be accessible from a data base stored on the instructor workstation or from a central data base over a LAN. The DLS receive sites shall always be able to receive an initialization command from the instructor; hence, any activities on the student workstation shall be interrupted and the DL software shall be started.

The DLS shall provide a function that allows instructors to either determine how many students are participating in a DL session or identify students on an individual basis. In the former case, the instructor shall allow students to send a signal from their input device and the system shall tabulate attendance from all participating sites. Attendance data shall be displayed in a Roster display window. In the latter case, students shall enter an identification number during roster generation. The DLS shall tabulate the attendance entries and correlate student names to ID numbers, which are retrieved from a central or local database. The DLS shall display the total number of students at receive sites, as well as the names of the students when requested.

The DLS shall provide a capability for instructors to determine how well students are grasping the material; this shall be accomplished using test and question generation functions. The question function shall allow an instructor to ask true/false or multiple choice questions at any point during the DL session (ad-hoc feature). The DLS shall provide real time display of results to the instructor and receive sites. The test function shall provide an encrypted, password protected text editor for creating test questions. Test questions and answers shall be capable of being accessed from a central or remote database. At the conclusion of a test, the instructor shall have the ability to view student responses on an individual basis and perform statistical analysis on the results.

Student interaction with instructors shall be supported through the use of a hardware or software emulated SRUs. The emulated unit, which is used to emulate an input device on a workstation, shall support all the functions provided by the hardware device. Keypad types used at receive sites shall be transparent to the instructor. SRUs shall provide help keys to allow students to signal the instructor when help is needed. The DLS shall notify the instructor of the requested help by displaying a message box on the instructor station identifying the receive site location. When the instructor acknowledges the help request, the site name shall be inserted in the Help request list box. Next, the instructor shall have the option of canceling the help request or opening up a dialogue by enabling the receive site voice equipment. While a receive site is pending a response to its help requests, the DLS shall disable all other SRU help request keys at that receive site.

The DLS shall monitor the status of all receive sites configured into the DL session. In most instances, system monitoring shall be automatic with message transmissions to the instructor when receive sites become inactive or when equipment failures are detected. The DLS shall

read a system initialization table to determine the interval cycle for monitoring receive sites. This table shall be encrypted and password protected, and shall be accessed from a local or remote database. When a receive site is identified as inactive, the instructor shall be notified and the Roster window shall be updated to identify the site as inactive. The DLS shall notify the instructor when the site becomes active and update the Roster window to indicate the site is on-line.

Student

The DLS shall provide methods for the student to participate in a DL session using hardware keypads and a large screen projection system or using a PC with emulated keypads. Students shall have the ability to register into classes, answer questions, request assistance, speak to the instructor and take tests. Microphone and speaker systems shall be enabled by the DLS when the instructor opens communication with a student and be disabled when the instructor terminates the conversation.

Administrator

Although the DLS shall be designed so one person could operate the entire network, the system shall provide system console support for maintaining LAN, terrestrial and satellite equipment. Administrators shall have the ability to run diagnostics tests on all DL equipment. Administrators shall also have authority to override the existing communication equipment and enter new parameters. All errors detected shall be displayed on the administrator console and, depending on the error, shall halt execution of the DL control software. Access to equipment parameters shall be secured using a password and encryption. The administrator shall be provided an edit function to enter and change equipment parameters. The edit function shall also be used to enter receive site configurations. Equipment and configuration information shall be accessed from a local or remote database.

SYSTEM ARCHITECTURE

The DL software architecture is shown in Figure 1. The architecture is partitioned into four system components: Instructor Interface Control System (IICS), Student Response Control System (SRCS), Receive Site Control System (RSCS), and Network Control System (NCS). System components communicate with each other over LAN, satellite and terrestrial interfaces using TCP/IP, a De-Facto communication protocol. Referring to Figure 2, data flow within the DLS starts with instructor commands and status transmissions between the IICS and NCS. The IICS component is responsible for accepting commands from the instructor and displaying network status. Instructors will also use the IICS component to define classroom receive site configuration; initialize receive sites, display test and question results; enable receive site dialogue; and access classroom rosters. Rosters and tests can be accessed from a central repository over the LAN or WAN interfaces.

The NCS component will control terrestrial and satellite equipment and will be the focal point for information passed between the instructor and receive sites. The NCS sends commands and receives status from all receive sites participating in the classroom lecture. The RSCS is responsible for communication with the broadcast site NCS and controlling data flows to and from the SRUs. If the SRUs are hardware devices, the RSCS will have a direct connection to the SRUs. However, if the SRUs are emulated on a workstation, then the RSCS will communicate with a SRCS to control data flow to and from the student. The RSCS and SRCS will be connected on a LAN at the receive site and will utilize TCP/IP as the LAN communication protocol.

Specific features within the four system components will be offered as software modules. Notice that software modules do not have direct communication links to each other (Figure 1). Modules communicate with the executive program using a standardized inter-module communication interface. It is this standard interface that will allow software upgrades to existing platforms with minimal replacement of existing software or hardware.

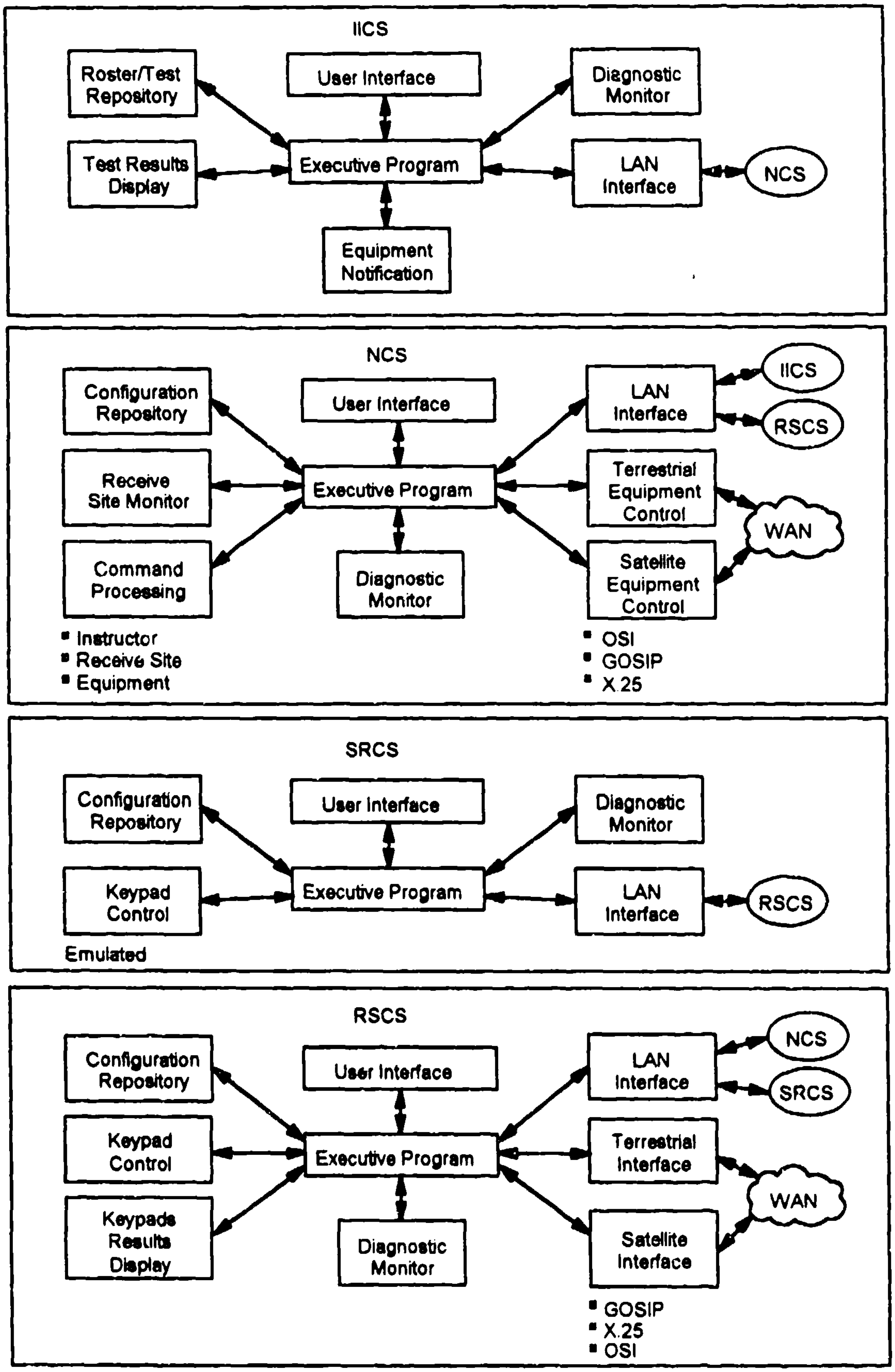


Figure 1. DLS Control System

06/93000596/F158

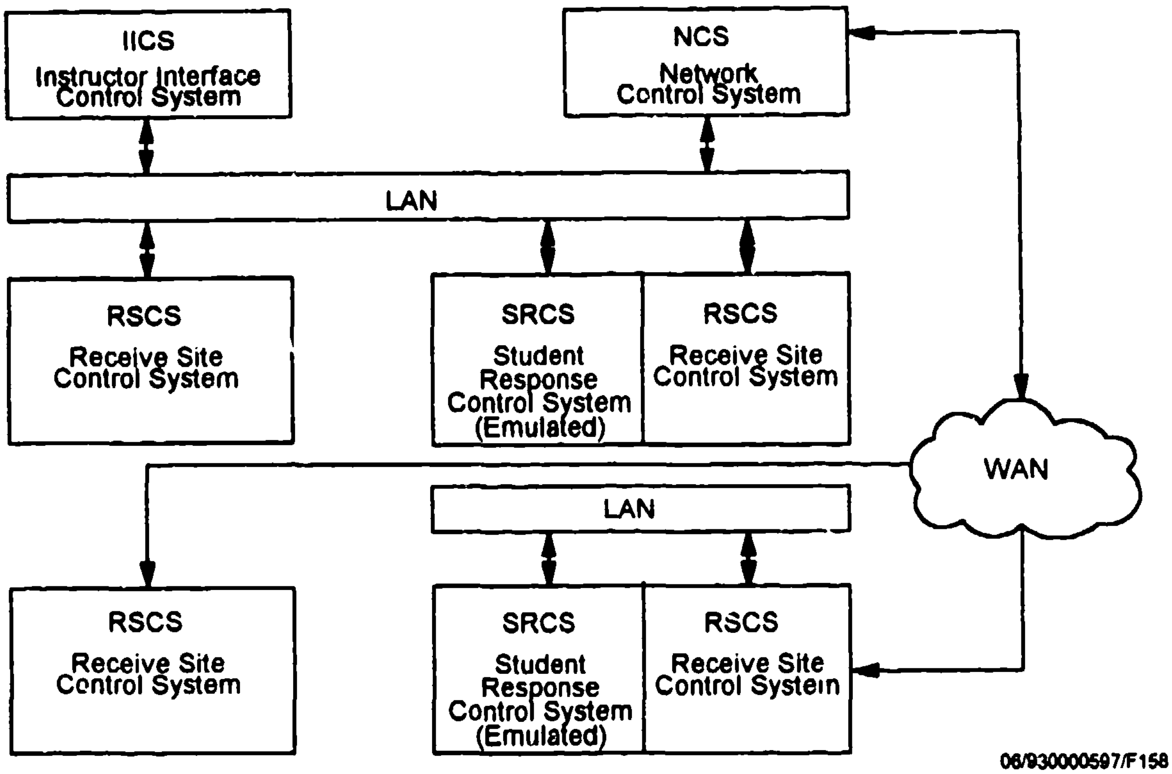


Figure 2. DLS Data Flow

The DLS will utilize the SQL interface, along with the DCE Remote Procedure Calls (RPC), to access information from file/database servers. This architecture can also support the CALS initiative by adding software modules that will transform information into CALS compliant formats before data is transferred to the central repository.

Notice that separate modules are used to control communication over WAN and LAN interfaces. By developing communication interface software as functional modules, the DLS will be able to utilize software that complies with the full GOSIP/OSI stack; portions of the GOSIP/OSI stack; or utilize an internal military communication standard.

The Executive program will control the flow of information throughout a DLS component. Specifically, the Executive program will: read a configuration table; select the appropriate interface drivers; control data transmissions between the DLS components; and control information transmissions between internal modules. The power and flexibility of the executive program is dependent on the type of operating system selected. Although the best choice for an operating system is a POSIX compliant system, current POSIX compliant systems are limited in multimedia support compared to DOS, DOS/Windows and OS/2 operating systems. The NCS will require a multitasking operating system due to timing constraints and the amount of events that have to be managed. Although it is recommended to use a multitasking operating system in the RSCS, IICS and SRCS components, the components can provide all the functional capabilities described in this paper using a single tasking operating system.

DLS Control System

The control system will encompass all the software and hardware that manages the flow of data between the Instructor and the student. As shown in Figure 1, within each component are modules that perform specific functions. Because many of the component modules perform the same functions, only the NCS and RSCS components will be discussed in detail.

The NCS component is responsible for the operation of the DLS. Specifically, the NCS Executive program will be responsible for initializing the DLS network; monitoring receive sites; and processing instructor and administrator commands. The NCS Executive program will utilize a round-robin method to perform all of its tasks. Instructor commands, incoming messages, outgoing status and hardware errors are constantly checked. There will be instances when the round-robin cycle is interrupted, such as when an catastrophic hardware error occurs or when an instructor command is detected. By using this round-robin and prioritization technique, the NCS can guarantee system performance in executing instructor commands and processing receive site responses.

The NCS Executive program utilizes the other modules to communicate with receive sites, equipment and data bases. The Configuration Repository module will be responsible for communicating with data bases and allowing instructors to build configuration tables. At initialization, the Executive program will request initialization information from the Repository module and will initialize the system. At this point, the Receive Site Monitor module will begin sending commands to the receive sites to determine their status (active or inactive). The Executive program will use the configuration data obtained from the Configuration Repository module to determine what interface module to use (LAN, Satellite and/or terrestrial). Receive site responses will be handled by the appropriate interface module (LAN, Satellite or terrestrial) and will be forwarded to the Receive Site Monitor module. A list of active and inactive sites will then be generated and maintained. Throughout the distance learning session the Receive Site Monitor module will monitor the status of receive sites by issuing polls on a constant cycle. Receive sites will be considered inactive if they do not respond to a predetermined amount of poll requests. When receive sites move from inactive to active, the Receive Site Monitor module will update its status list and notify the IICS and NCS components.

Incoming commands from the NCS and IICS components are processed by the Command Processing module. When commands are received, they will be prioritized and inserted on a queue. High priority commands, such as equipments errors, will be processed immediately. The Command processing module will terminate a DLS session if a severe equipment error is detected. The Command processing module also formats and packetizes instructor commands for transmission over the three interfaces.

The Diagnostic Monitor module will monitor internal module communication and detect software errors. Errors detected will be logged in a central or remote database for analysis purposes. When software errors are detected, the Diagnostic module will notify the IICS or NCS components.

Communication equipment will be controlled by the LAN, Satellite and Terrestrial communication modules. These modules also packetize and format data to the standard specified in the configuration table (i.e., TCP/IP and GOSIP).

The RSCS will control all data flows between the NCS component and the SRUs. Poll and status commands are processed by the RSCS and, depending on the receive site return interface, will be transmitted back to the broadcast site upon receipt of a request for status poll or will be transmitted immediately. Keypad commands such as enable/disable keypad and start/stop help request polling will be processed by the Keypad Control module. The Keypad Control module will poll keypads at least twice the speed of the NCS receive site poll to ensure help requests are detected within one NCS poll cycle.

The RSCS will operate as a broadcast site processor or a remote site processor. In remote site mode, the RSCS will control hardware SRU devices or communicate over the LAN with the SRCSs to control the display of software emulated SRUs. In broadcast mode, the RSCS has the above capabilities and will also collect keypad results from the NCS and display the results on its monitor using the Keypad Results Display module. The RSCS will only operate in this mode if it is attached to the NCS over a LAN interface. This capability will allow the instructor to broadcast from any LAN attached classroom.

All of the remaining modules in the RSCS provide the same functions as described in the NCS section above.

IBM has performed tests on the performance of a DLS that has been delivered to a customer. The system is very similar to the DLS being proposed in this paper. The NCS component was developed on a 386 machine running at 25Mhz. We were able to monitor 10 receive sites within a three second poll interval and 40 receive sites within a ten second poll interval. When an instructor requested collection of student keypad data from 40 sites, the results were displayed within 10 seconds. Of course, the performance is dependent on WAN speeds, as well as processor speeds and return communication interfaces. However, this test showed that the round-robin technique can provide sufficient performance when large numbers of receive sites are configured into a DL network.

User Interface Design

Targeted users of the DLS are not necessarily computer experts or even computer-literate. Therefore, our interface design focuses on specific human factors such as: consistent button layouts; consistent message formats; system guided user selections; Common User Access (CUA)/ Graphical User Interface (GUI); touch, mouse, keyboard input capabilities; and an overall friendly user interface. Three interfaces are discussed in the following paragraphs: instructor, student and administrator.

Instructor

The IICS will be the instructor's main interface to the DLS. A windowed environment will be developed to allow the instructor to easily toggle between a DL session and other activities being performed on the workstation. The graphical buttons provided to allow the instructor to have complete control of a DL session are Receive Site Setup, Start/Stop Roster Generation, Ask/End Question, Start/Stop Test, Enable/Disable Audio Dialogue, Disregard Help Requests, Classroom Status and End Class. Because many functions must precede others, the buttons will gray in and out as each function is performed. The user interface will guide the instructors through the DL control features. Therefore, an instructor can not inadvertently cause a system crash during a DL session. Another capability that will be provided is an ability to toggle functions using one button. This capability helps to logically group functions and decreases confusion due to a cluttered screen. For example, the Ask Question button toggles to End Question, and vice versa. Message boxes will be used to notify the instructor when a student has a question; when errors are detected in interface equipment; or when a receive site becomes inactive due to transmission problems. List boxes will be used to store help requests and receive site roster data. This information is visible on the screen at all times. List and message boxes will have a consistent format for all types of status and error information being displayed to the instructor.

Student (Emulated Input Device)

The main responsibility of the SRUs will be to provide a graphical user interface to students for inputting data, requesting help and viewing a lecture. This interface will also be in a windowed environment, allowing the user to toggle back and forth between a DL session, CBT application or any other type of computer application. A graphical representation of an input device will be provided, which will be presented in a separate window. The device will have a display area to allow the student to view the information being entered. Student input devices will support both numeric and text entries. The display area will scroll up and down when text information is entered. Student input devices will be activated by the instructor (ask questions and roster generation) or by the student (request help). When the input device is not active, it will appear as an icon on the screen. Students will view lectures in a separate, scalable window. Furthermore, students will have the ability to move the video window anywhere on the screen and change sizes from 1/4 screen to full screen. The interface will also allow the student to toggle between the input window and video window during a DL session. Similar to the input device window, the video window will be controlled by the instructor and student. Touch, keyboard and mouse inputs will be supported.

Administrator

The NCS will be responsible for controlling the DL interface equipment and providing a windowed user GUI to administrators for monitoring equipment, initializing equipment and displaying messages. Similar to the IISC interface, buttons will be provided to allow instructors to select specific options. Certain

functions will be password protected. Message boxes will be provided to display parameter variables and error messages. List boxes will be used to allow the administrator to view and select from a range of equipment parameters.

SYSTEM IMPLEMENTATION COSTS

The cost associated with developing a DLS having all of the capabilities described above can be insignificant if the organization has an existing LAN based infrastructure. The DLS software could be loaded on the existing PCs and multimedia cards could be added to provide video and audio capabilities. A low end solution would be to provide distance learning across a building using an analog network to transfer modulated audio and video signals. There are technologies available that will allow the transmission of analog audio and video, along with baseband data, over existing LAN cables.

If receive sites are large distances from the broadcast location, video and audio transmission equipment will be needed. Cost trade-offs between terrestrial and satellite transmissions should be performed. As a simple rule of thumb, if the number of receive sites is more than four, it may be cheaper to use the satellite system. This formula may not be true in the near future with advancement being made in high speed digital communication services over land lines. Carrier charges will have to be added to life cycle costs. Also, charges for purchases of video compression, satellite and echo cancellation equipment will increase costs.

DLSs can only be justified if the system can effectively be used to train students and its cost is within the existing training budget. Cost justifications for DLSs involve an analysis of how conventional training is being conducted. Costs such as instructor travel, student travel, building costs, room and board, and lost time on the job should be included in cost estimates. Furthermore, the number of student being trained, number of instructors and future training requirements should also be factored into the cost estimate.

The proposed architecture will allow training organizations to better justify a DLS because the system operates in an existing client/server environment. Every user on the network can be trained without buying expensive communication and compression equipment. As the existing infrastructure grows to support more users, the DLS transmission capabilities grow also. For example, when high speed communication networks become available at cost effective prices, the DLS can be upgraded to support digital transmissions of video and audio over the network. This upgrade does not affect the DLS because the compression equipment is not tied to the software platform.

CONCLUSIONS

The development of a DLS requires a full understanding of standards and future technologies. The explosion in utilization of PCs in the home and work place has lead to an environment where PCs will be the primary device used for training and education. Therefore, it is imperative for training developers to provide solutions that will allow organizations to train people in their office and home environments. However developing DL training solutions, develope. must provide systems that support a seamless integration into existing client/server environments and allows growth without major changes to existing hardware and software. The system must also allow instructors to use conventional stand-up lecture teaching methods so that student comprehension/retention is not compromised by the new system. The implementation of our proposed system satisfies the requirements described above because the system is nothing more than a set of client/server applications running in a networked environment. The software architecture is designed to adhere to standards and, therefore, supports an Open System environment. IBM has developed and delivered DL solutions that utilized portions of the proposed architecture.

CLASSIFICATION OF ELECTRONIC CLASSROOMS FOR USE OF INSTRUCTIONAL PROTOCOLS

William A. Platt Ivor K. Davies James R McConville
Stephen J. Guynn Gary Orwig Charles G. Bollmann
Ron Bower Marvin Smith

ABSTRACT

Recent developments in electronics and computer science have been so dramatic that their incorporation into classroom design has caused the term electronic classroom to come into wide use. The purpose of this paper is to explore basic design and training issues for the electronic classroom and to isolate effective practices where they can be identified from experience in the field. To this end, several training sites were investigated to review the teaching and learning that were taking place. Experienced-Derived models of classroom procedure were developed for each situation. A system of notation was created that captured classroom interaction, media use, and personal control. A design classification was used to formulate protocols that fit each situation. The protocols covered steps needed to implement each strategy by incorporating type and frequency of interaction, information source input, communication patterns, locus of control, and type of feedback. The instructional protocols are sets of operating procedures for instructors to use in planning and executing instruction in electronic classrooms depending on the type of electronic classroom. The protocols were devised as a practical extension of learning theory modified by field experience.

ABOUT THE AUTHORS

Dr. William A. Platt holds a Ph.D. in Instructional System Technology from Indiana University. He is a training consultant to business and government. Dr. Ivor K. Davies is Professor of Education and Instructional Technology at Indiana University. Dr. Stephen Guynn is an Evaluation Specialist with Indianapolis Public School System, Dr. Guynn holds a Ph.D in Instructional Technology from Indiana University. Dr. James R. McConville manages the Training Systems Engineering Department at Loral Defense Systems, Akron. His duties include Research and Development of Advanced Simulation Systems. Dr. McConville holds the Ed.D. in Instructional Technology from Indiana University. Dr. Gary Orwig is Professor of Instructional Systems at University of Central Florida. Dr. Charles G. Bollman is the head of instructional services for the FBI Academy, LTC Ron Bower is operations coordinator for Marine Corps Service Support Schools. Doctoral candidate Marvin Smith teaches air control at Embry Riddle Aeronautical University.

CLASSIFICATION OF ELECTRONIC CLASSROOMS FOR USE OF INSTRUCTIONAL PROTOCOLS

William A. Platt Ivor K. Davies James R. McConville
Stephen Guynn Gary Orwig Charles G. Bollmann
Ron Bower Marvin Smith

INTRODUCTION

Instructional Needs

Tremendous pressures in the current education and training environment and the need to do more with less financial resource have added to the frustration of the educational community as it seeks effective solutions that appear just out of reach. Over a period of years various "hard" and "soft" technologies have presented hopeful educators with new sets of tools designed to increase the efficiency or effectiveness of the teaching-learning process. The latest of these is the electronic classroom. [1,2,3] Like other technological innovations such as programmed instruction, instructional television, and computer-aided instruction, the concept is marked by mixed results and multiple definitions. This situation has refocused attention toward the reinvention of tools, processes, and values that had been taken for granted as fixed aspects of the educational setting. The classroom which was by in large a product of the industrial age, must now be redefined in terms of the information age. The following definition was used to guide an investigation of several electronic classroom sites involved in either research or the delivery of instructional programs.

Definition of The Electronic Classroom

An electronic classroom is a computer-supported classroom designed to facilitate student learning by enhancing the interactivity between students and subject matter through the use of electronic media.

There has been a fundamental shift in vision that goes with this new definition of the classroom with the learner elevated to a position of greater prominence. New organizational structures and new practices are emerging to fill the space created as our culture copes with a period of transition. Several principles underscore the changing situation: (1) Classroom organization will be designed around the student not the teacher. (2) Learners will do more of the work of instruction. (3) More instructional decisionmaking will be in the hands of the students. (4) Students will perform whole tasks and integrate their learning to work situations rather than study isolated fragments fed to them by the teacher. (5) the organizational structure of the teaching learning hierarchy will become flatter and broader. And (6) Computers will become the central tool of the classroom. This last point is perhaps the most visible characteristic of the design changes being considered for electronic classrooms. Computers will have three uses: (1) control of media at the instructor console, (2) delivery of individualized or group instruction, and (3) response evaluation and retention of instructional knowledge in so-called "smart systems." All of this will be marked by greater interactivity in both instructional strategy extremes -- that is, group lecture on the one hand and individualized self study on the other.

Purpose

The purpose of this paper is to provide benchmarks to help determine electronic classroom design and employment strategy in actual teaching and research settings. Several field sites were studied. The sites were chosen not as examples of perfect solutions, but as locations where the leadership was committed to meeting the challenge of classroom redesign. A conceptual framework was used to capture the range of use and to classify the design of electronic classrooms. Each cell of the classification matrix represented a potentially different challenge. A protocol model and notation system was used as a means to capture the critical success factors needed for each of the situations found in the field. To this end a protocol model was created. The protocols provide rules that will optimize interactivity of the instructional delivery process to increase classroom learning in a predictable way.

Reinvention of the Classroom

The transition period will last for several years with both traditional and experimental solutions existing side by side. Traditional academic classes are marked by an implicit assumption that learning is spread over an extended period involving study, homework, and gradual change in behavior. The academic model is, however, less attractive to the military which places great emphasis on learning during the instructional period. To military instructors it makes sense to have learning take place while they are able to guide it and to observe progress. Focusing on the classroom as a location for learning has stimulated the use of technology and the role of the student and the instructor in classroom learning. Some technology is focused at reducing the workload of instructors, while other technology is focused on giving them more control. The role of the student is not always the same in each situation. Different degrees of interactivity are appropriate at different times and for different subjects. Without a specific design and clear training objective, these issues can not be resolved. Therefore a central theme of this paper will be to resolve these issues at the protocol level.

Instructional Protocols as a Function of Electronic Technology

At least ten to fifteen years of experience with electronic media have produced considerable data on what works and what does not. That experience must be translated into models of value to the practitioner in the field. The heart of the matter is the formulation of training strategy that builds upon behavioral science and is specifically designed to take advantage of the functional capability of current electronic technology. Electronic media present a large array of functional possibilities, but it is the educational value of the individual medium that must be identified and codified in practice.

Design Issues

There are design issues which are best treated in specific situations. These include: (1) type of computer architecture, (2) instructor control versus student control, (3) contingent behavior and learner reinforcement, (4) type of courseware, (5) the use of authoring systems, and (6) the degree of interactivity and feedback. These issues represent design problems that have great potential to effect the critical success factors identified in each protocol.

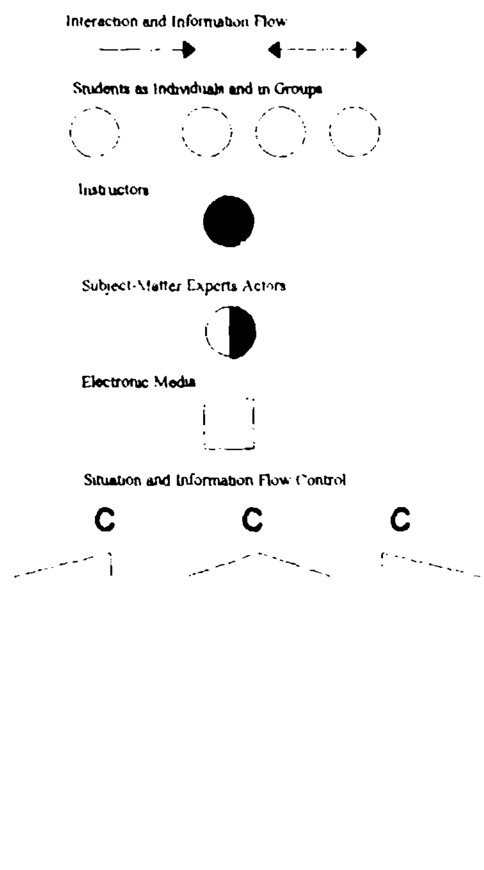
CLASSIFICATION FRAMEWORK

It is important that a classification that will be used in a period of transition presents a framework that includes elements of both the

past and the future. Three issues bridge the gradual shift in emphasis over time. These then provide three critical dimensions of design: (1) presentation mode, (2) generality of purpose, and (3) degree of conferencing capability. Figure 1 identifies the logical possibilities based on these three characteristics. Theater and carrel modes refer to the source of the information being presented to the learners. In theater mode all attention is directed to the front of the room. In carrel mode each student has an information source at a student station. Some classrooms are set up with both and are, therefore, mixed mode. Classrooms are also either dedicated to a specific function or are open for general use. Dedicated instructional classrooms such as simulation laboratories often present special problems to the instructor and student. These problems must be addressed in the instructional strategy in use. General purpose classrooms offer the possibility of transferring effective instructional technique from one class to another. Protocols should reflect this difference. Classrooms also differ in conferencing capability. As used here conferencing refers to any means of interactive electronic communication. Examples are the networking found in distance education, simple intercoms, responder systems, and interactive bulletin boards using computers. Conferencing generally implies two-way communication, but one-way response systems are included under the concept used in this classification. By categorizing classrooms as to presentation mode and generality of use, a matrix of six cells was created. Each of the cells can in turn be classified as having or not having conferencing capabilities for students to interact with each other or the instructor, both inside and outside of the room. A system of notation can capture other aspects of the instructional setting within each of the cells of the matrix.

INSTRUCTIONAL NOTATION

A simple notation was created to depict the source of information, the nature of the interactivity the players involved, and the control of the situation. Appendix A depicts the classroom design for each of the field settings. Notation has been added to diagram the teaching learning situation in each of the classrooms depicted. The notation has also been added to each of the field descriptions below. The notation includes symbols for teachers, students, subject-matter experts / actors, communication flow, electronic media, and control. The symbolic representation is as follows:



	Multiple Use		Dedicated Use	
Theater Mode	Conf	Non-Conf	Conf	Non-Conf
Carrel Mode	Conf	Non-Conf	Conf	Non-Conf
Mixed Mode	Conf	Non-Conf	Conf	Non-Conf

Figure 1 Classroom Matrix

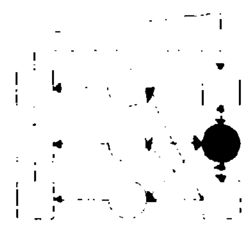
EXPERIENCE AT THE SITES

Differences in usage and design were found in the field settings. The seven sites were however, all actively looking for new ways to improve instruction. Each site was investigated to review the use of classroom technology and to identify problems that have forced designers to cope with the instructional delivery problem in different ways. Where possible, critical success factors were identified.

FBI Academy

One of the earliest full scale attempts to increase the use of technology in the classroom occurred at the FBI academy at Quantico, VA. Classrooms featured a response system, rear screen and front screen projection, and networked video distributed on call from a central point to classrooms. The student and instructor stations are depicted in Appendix A. The Instructional control and interaction is depicted as follows:

C

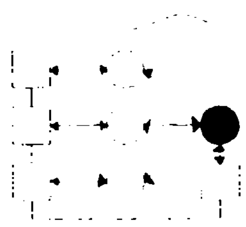


Much has been learned as a result of years of experience. The importance of training instructors to operate the equipment was learned in regard to the responder system. The use of responder systems also was found to be dependent upon greater instructor preparation time. Use of the responder system for testing proved impractical because of the level of technology which was available when the system was installed in 1972.

Marine Corps Base Camp Lejeune MCSSS

One innovative use of computers in electronic classrooms is at the Marine Corps Service Support School, Camp Lejeune, NC. Stimulated by the need to train financial managers to use computers, classrooms were designed with networked computers at each student position. The instructional interaction and control is depicted as follows:

C

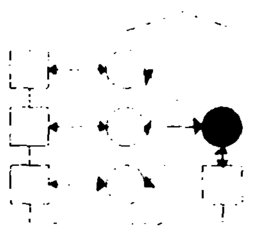


Instructors found that in addition to teaching students how to use the computers, the computers also offered a means of delivering a wide variety of instruction. By networking monitors and using a single instructor console to generate a signal,

instructors were able to use presentation programs to teach lectures in other schools including Motor Transport, Food Services, Supply, and Administrative schools. Instructors were able to take charge of their own presentation production which resulted in faster updates and individual initiative in presentation design.

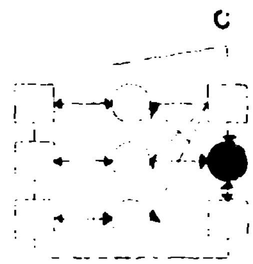
Indiana University

The School of Education in conjunction with the Division of Instructional Technology conducted experimentation in the 1970s with a classroom that had centralized control at an instructor console, rear screen, mobile use of TV and video taping, rear screen projection, and a responder system. The school of education has benefited from the early work in classroom design. A new building features networked classrooms including two computer classrooms. One of these is used as an open laboratory, the other as a scheduled classroom. Emphasis has been placed on the learning of computer tool skills to enhance learning in other areas. A wide-area network facilitates data transfer and provides the infrastructure needed to originate distance-education programs. A typical interaction and control configuration is depicted here:



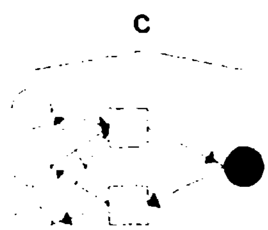
University of Central Florida

The University of Central Florida and the Institute for Simulation and Training use an electronic classroom to conduct research on the design, development, and delivery of training. Current research topics address the use of artificial intelligence (AI) and expert systems as tools to facilitate ways to increase local production of individualized instruction. The aim of the research is to make a training system more responsive to the needs of classroom users. A typical configuration follows:



Indianapolis Public Schools

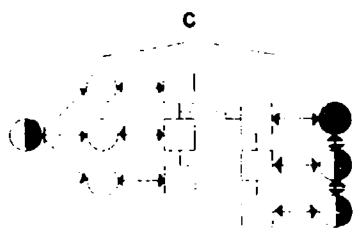
Prototype development is underway to link different settings using interactive electronic technology. The "worlds largest fiber optic cable network" [3] has been established linking three high schools to teachers at Indiana University. High school classrooms are equipped with cameras and large screen TV monitors for two-way television transmission in real time interaction between teachers in one locale and students in another. The configuration is:



Embry Riddle Aeronautical University

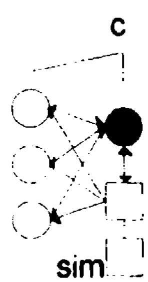
The development of a classroom designed to simulate an air control facility has recently been initiated. This facility uses consoles approximating the new air control consoles in general use. Features have been added to prepare the students for changes in the air control system brought about by recent upgrades to the FAA system. The system is capable of simulating an air control screen and presenting computer-based instructional sequences on the same screen. Live player

simulation inputs are provided by pilots in a separate room equipped to provide the required inputs for voice communication and aircraft flight. Instructors can play the role of aircraft pilots who can interact in real time with the students. Some instructors act as coaches and provide "over the shoulder" help to the students when needed. The configuration follows:



Loral Corporation

Loral Defense Systems, Akron, OH has undertaken the development of several prototype classrooms that will link classroom instruction with simulators in real time. A demonstration and briefing room is equipped with video projection and multimedia control console at the podium. Loral also has developed simulator suites where computer-based training can be linked to real time simulations. Loral is developing classroom concepts using the technique of "rapid prototyping" in the manner suggested by Tripp and Bichelmeyer. The configuration and interaction diagram is:



INSTRUCTIONAL PROTOCOLS

The protocols are based upon the idea that classroom teaching strategy: (1) should take advantage of the design of the classroom in which the teaching is taking place, (2) should be aimed at achieving learning objectives by taking advantage of the options offered, and (3) should identify critical success factors that are common to all designs as well as those that are unique to a specific situation. Therefore the protocols are organized as rules for using each of the classroom configurations included in the classification matrix. By focusing on one cell at a time, the clarity of purpose and instructional intent is increased.

Protocol model

A protocol is a set of instructions that when followed will assure that instruction follows a pattern that has proved effective. The protocol model is divided into components which includes the preparation of instructional material in the format and medium needed for delivery in the electronic classroom. The model used here contains the following components: (1) Preparation, (2) Setup, (3) Delivery and Rehearsal, (4) Interactive Dialogue and Pattern, (5) Control and Evaluation Feedback, (6) Records, and (7) Closure. The key to all of the protocols is the way in which the learner is stimulated to be an active participant in the learning process. In some of the classroom applications a passive learner is more typically the case. For these situations the protocols change emphasis from delivery interaction to preparation of content that is organized to stimulate the learner to engage in active cognitive participation.

PROTOCOL EXAMPLE

The following protocol is identified by the three dimensions of the classification matrix. Each set includes a brief introduction and example that fits the set categories.

Mixed Mode, Multiple Use, Conference Type

This type electronic classroom is commonly used to present prepared lectures where the participants can work on individual materials during the instructional period and communicate with others inside and outside of the classroom. Presenters design the content to introduce a topic and then let the students practice at their own pace using equipment at the student station. Electronic podiums and response and communication systems are geared to monitor each student and to allow for individual and group communication The classrooms at the Financial Management School and the Personnel Administration School at Marine Corps Service Support Schools are of this type.

(1) Preparation. Plan the lesson or briefing using learning objectives or briefing points and prepare lesson content materials to support the objectives in presentation order. Particular attention is paid to the structure of the lesson/lecture. Content must be logical and teaching points used to lead into the practice sessions. Plan for classroom assistants to help the instructor. Identify teaching points to focus the attention and alert the audience to possible areas of difficulty with the material. Critical Factor: Lesson material in interactive format must be available at each student station.

(2) Setup. Check the equipment. Ensure that the media work. All items to be used together should be compatible. Ensure that on-line data bases are up and working. Critical Factor: Equipment check-out and testing, compatible hardware and software.

(3) Delivery / Rehearsal. Rehearse the delivery to gage timing and emphasis. Monitor student progress and use student examples to point out problems. Ensure that students are given time to practice correct procedure after making errors. Time the conference discussions in rehearsal. Research any unanticipated questions. List source references. Prepare follow up points and summary review points. Critical Factor: Timing and sequence.

(4) Interactive Dialogue and Pattern. Note the extent of audience participation in the conference mode, and guide audience interaction by showing examples to the class. Instructors can guide the student practice sessions by individual and group coaching. Critical Factor: Control, motivation and question structure; corrective instruction; practice and reinforcement.

(5) Control and Evaluation Feedback. Overall control rests with the instructor who can take over at any time. Learners have control over the pacing of their individual practice sessions. They can initiate communications to external file servers. Audience/Learner conferencing is moderated by the instructor who notes the quality and accuracy of audience response and provides feedback at the earliest point in the dialogue consistent with courtesy. When the evaluation feedback is provided on a subject that has clear right and wrong responses, rehearse the correct response after providing feedback on the error. Critical Factor: Timing, personal motivation, group norms.

(6) Records. All record keeping is performed jointly by instructor and students. Instructors record audience queries and make electronic notes at the instructor console. Students make entries into logs and records made available by the courseware. Critical Factors Student awareness of progress in relation to the course outline.

(7) Closure. Summary statements and direction for future study are made by the instructor who includes reference to any unanswered questions that have not been resolved during the class period. Critical Factors: Motivation, planning, concept reinforcement.

General and Specific Factors

Some critical factors can be divided into general and specific. The most consistent finding across field locations was that effective use of electronic classrooms was directly related to instructor training and preparedness. In contrast to this general factor, specific critical factors exist for specific settings and situations. For example the accurate ability to role play the part of a pilot is specific to the dedicated air control classroom.

RECOMMENDATIONS

Recommendations include specific actions and changes in point of view. It is necessary for instructional designers to comprehend the subtle and profound changes occurring as the information age replaces the industrial age. This change brings with it a tremendous opportunity. A new paradigm must replace the old concept of the classroom. Protocols can play a role in making instructor training more consistent and productive. But they are more important as a tool for change when they cause the reexamination of basic assumptions about students and teachers. Increasing the effectiveness of training for the instructional staff is perhaps the area of greatest potential for enhancing the effectiveness of electronic classrooms across all categories of the classification matrix. Protocols can be tailored to enhance the degree of skill and understanding of the instructional staff in applying instructional principles and in using equipment to advantage. Greater focus on the teaching and learning can occur converting an electronic technology into what Davies , has referred to as "performance based technology" where real gains can be made.

Protocol Design Hypothesis

This project hypothesized that inquiry into the history of the several sites would reveal a relationship between learner interactivity and the general success and longevity of the classroom design. In general this idea was supported - with one surprising exception. Interactive response systems (as presently designed) are underused. Several factors that have masked or moderated the relationship between interactivity of the design and the perceived success of the facility were identified. The first of these is the lack of a precise definition of interactivity. Second, is the swamping effect of subject matter organization and presentation over media modality. The third was the functional limitations of early equipment used for responder systems. Fourth, was the teacher training and skill at creating a motivating interactive learning environment.

Dedicated Research for In-service Settings

The field settings studied included some that emphasized pure research and some that carried on day-to-day instruction. Both are needed. One activity generates ideas. The other checks them against the reality of day-to-day practical stress. This latter point cannot be overstated. It is highly recommended that electronic classrooms be tested in actual use. The protocols represent a first attempt to compensate for limitations and to enhance the advantages of real settings by bringing attention to critical success factors. It is expected that protocols for each setting will be modified as they are used as the basis for planning instruction and training instructional staff. The protocols should be treated as a dynamic solution that needs to be continuously refined within each of the cells of the matrix to ensure optimal application.

Focus On Teaching and Learning Process

The point of having protocols is to focus more on the teaching and learning interactions and less on the equipment-oriented aspects of the electronics. Avoidance of future pitfalls for the use of electronic classrooms features will require a less intrusive equipment environment so that teaching and learning can prevail. This will become even more important as the role of data bases and networked resources becomes a more evident factor in instructional technology.

Specific Design Recommendations

Design solutions should promote active learning in an environment that includes corrective feedback, motivation feedback, and flexible media under student control. Responder systems have not worked well in the past but have great potential. The causes of the poor results relate to teacher training, instructional planning, and equipment limitations. It is recommended that interactive response systems be included in future designs, accompanied by adequate training of the instructional staff. Time must also be given to the preparation of the content materials for an interactive format. The equipment must be easy to use, reliable, and fast. Future designs should take advantage of advances in computer interactive networking. Multitasking environments will make it possible for students and instructors to engage in a dialogue while other programs (and other students) are performing other learning tasks.

Rapid Prototyping

One very useful practice is recommended for future electronic classroom design efforts aimed at meeting user needs on a timely basis. Rapid prototyping has been used successfully in industry to develop software. Trip and Bichelmeyer [6] have pointed out that the concept can be applied to instructional design strategy. Extension to classroom design also seems warranted. The cost and exasperation associated with planning new facility designs could be moderated by this approach, which was one of the methods reviewed by the Oklahoma University Instructional Technology Effectiveness Study. [7].

References

1. Chambers, J. A. et.al. (1992) The Learning Revolution: Electronic Classrooms.

2. Saylor, J. M. (1992) "NEEDS (The National Engineering Education Delivery System): If We Build It (According to Standards) They Will Come!" Proceedings of the ASIS Mid Year Meeting: pages 273-280.

3. Wilson, K. & Tally, W. (1991) Designing for Discovery: Interactive Multimedia Learning Environments at Bank Street College, Technical Report No 15, Center for Technology in Education, New York, NY.

4. INFO (1992) Office of the Superintendent, Indianapolis Public Schools, Novemger 6, 1992.

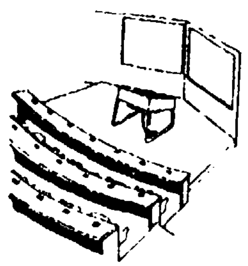
5. Davies, Ivor K. (1993) "Performance Technology." Lecture notes provided in personal correspondence.

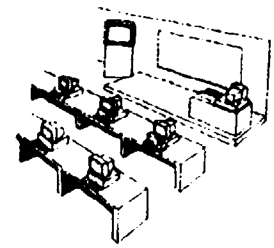
6. Tripp, Steven D, & Barbara Bichelmeyer (1992) "Rapid Prototyping: An Alternative Instructional Design Strategy," ETR&D Vol.38, No.1 pp 31-44 ISSN 1042-1629 3.

7. Skill Dynamics: IBM (1992) Paper 5 Fast-Track Instructional Design," Oklahoma University Instructional Technology Effectiveness Study.

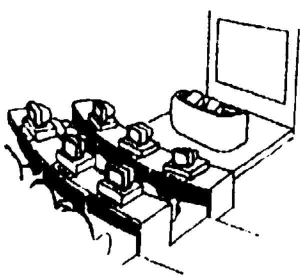
APPENDIX A
Electronic Classroom Configurations

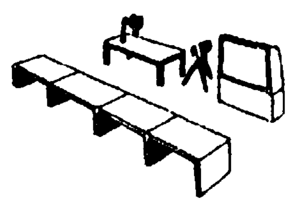
Left: FBI Academy: Theater Mode, Multipurpose Use, Conference Type. Right: Marine Corps Service Support Schools and Indiana University: Mixed Mode, Multipurpose Use, Conference Type.



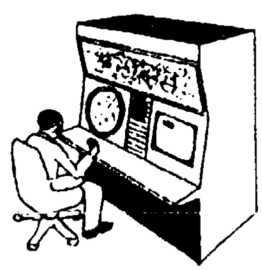


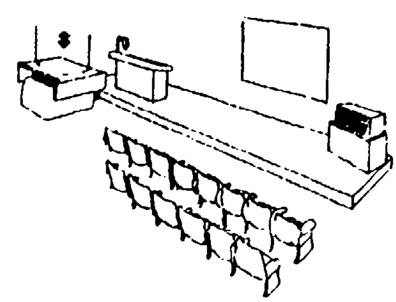
Left: University of Central Florida: Mixed Mode, Multipurpose Use, Non Conference Type. Right: Indianapolis Publid Schools: Theater Mode, Multipurpose Use, Conference Type.





Left: Embry Riddle Aeronautical University: Carrel Mode, Dedicated Use, Conference Type. Right: Loral Defense Systems Akron: Theater Mode, Multipurpose Use, Non Conference Type.





CONCURRENT DEVELOPMENT: BOON OR BANE?
AN ISD PERSPECTIVE

J.D. Jared
Boeing Defense and Space Group
Seattle, Washington

ABSTRACT

This paper focuses on the problems and adaptations required for the instructional systems development process to support the concurrent development of weapon systems and training systems. Government contracts mandate the use of the instructional system development process to develop training systems and also requires the concurrent development of training systems along with the weapon system. The advantages of concurrent development are obvious, such as having a training system delivered at the same time and in the same configuration as the weapon system first article. However, the concurrent development requirement constrains the options available to the training system developer. In the course of developing the analysis procedures for identifying training requirements of a major weapon system, considerable flexibility was required for the analysts and designers. What appeared to be a relatively straight-forward analysis and development approach, in practice, required a significant number of modifications, including procedures and software tools used in the front-end analysis. Specifically, the criteria used to select procedures and analysis models, primarily train/no-train and media selection, were driven by the concurrent development requirements. The program schedule, availability of design data, and analyst capabilities also had to be considered. A major concern for the training device developers is the changes which occur in the weapon system as it evolves. Changes are a critical and costly issue that must be addressed from the beginning. When concurrent development requirements are applied to new weapon system programs, there is a need for a tailored ISD process that sustains analytical integrity and supports the media developers.

ABOUT THE AUTHOR

J.D. Jared is an experienced training systems analyst in Boeing's Defense and Space Group's Military Airplane Division. He received a BA degree in Communications from the University of Washington, a BS degree in Computer Information Systems from City University, and an MBA from Harvard University. He has an extensive background in the development and management of complex military training programs. He is currently the lead for training technology on the F-22 weapon system development program.

CONCURRENT DEVELOPMENT: BOON OR BANE? AN ISD PERSPECTIVE

J.D. Jared
Boeing Defense and Space Group
Seattle, Washington

INTRODUCTION

In recent major procurements, the Government has required the development of training systems concurrent with the development of weapon systems and has mandated the use of the instructional system development (ISD) process. Specifically, new procurements have a requirement to field the training system concurrent with the delivery of the first operational article of the weapon system. This paper focuses upon the interaction and modifications to the ISD process needed to develop the training system in parallel with the weapon system. In most new programs, there are periodic changes in the weapon system program which have significant impacts on training. This paper discusses general requirements for training and the weapon system, then concludes with the specific changes made for the new training system.

Background

New weapon system procurements have tasked the prime contractor with the responsibility for training system development. Innovative procurement actions have been taken to ensure the highest quality product is developed for the lowest cost. These new weapon systems are being developed through the use of integrated product development teams (IPDTs).

Training Program Requirements

The new weapon system program discussed in this paper is currently in the engineering and manufacturing development (EMD) phase. A totally integrated training system is being planned and developed. The new training system consists of an Operator Training System (OTS), a Maintenance Training System (MTS), a Depot Training System (DTS), a Training Management System (TMS), and a Training System Support Center (TSSC). The size and demanding requirements of the new weapon system poses unique challenges for the training community, especially for those involved in the instructional system design process. The training system is being developed in five locations, including the engine manufacturer.

The goal is to field a training system concurrent with the first operational article, and in the same configuration. The mission analyses, human factors analyses, and the logistics support analysis were not completed prior to the start of the ISD process. Therefore, the source data available to the analysts was limited at the beginning of the program.

FRONT-END ANALYSIS

Major decisions for the development of the training system were planned at the early part of the program, sometimes referred to as the "front-end analysis" stage of the ISD process. The tasks to be trained and media used in training are determined during this part of the EMD Phase.

The training team's ISD process refers to MIL-STD-1379D as a guide. The ISD decision aiding models were not specified as part of the program contract, as the training program is to be developed using "best commercial practices."

Train/No-Train Models

Identification of the system operation, maintenance and support requirements is not a trivial undertaking, as this, and related information, feeds the train/no-train decision process. The train/no-train (task selection) process is critical to the development of the total

training system. This is the first major decision. Selection of the appropriate model to aid the analysts can enhance the effectiveness and efficiency of the decision-making process. There are a number of analysis models available to support the training analyst in deciding whether a task should be trained or not trained (see Table 1). Each model requires a different set of input data and have been validated on previous programs.

Table 1. Train/No-Train Models

MODEL	LEVEL OF OPERATION
EARLY COMPARABILITY ANALYSIS (ECA)	TASK
DIFFICULTY, IMPORTANCE, AND FREQUENCY (DIF)	TASK
4 FACTOR	TASK
8 FACTOR	TASK
3306 TES STAM	TASK ELEMENT SKILL/KNOWLEDGE

Media Selection Models

The selection of media for a training system is at best a controversial process due to the perceived costs involved. Like the train/no-train analysis, the input data is important to the process and its results. There are a number of media selection models to be considered (see Table 2). Each of the models have varying requirements for input data. This is a very important step in the ISD process, since it is the basis for further analyses leading to development of prime item development specifications.

Model Selection Criteria

The selection of the models to be used in the ISD process is dependent upon a number of factors. Supportability, tool availability, and analyst capabilities are criteria to consider when selecting the models. However, the availability of source data has been identified as one of the most critical for training programs involved in concurrent development.

A number of tools, such as the Joint Services Logistics Support Analysis Record/Instructional System Development Decision Support System (JS LSAR/ISD DSS) tool have the decision models contained within them. These tools and models have varying data input requirements that will often impact the selection of the models to be used. Even more frequently, it is the availability of the source data that will point the ISD analyst in the direction of the tool/model to be used.

In most new programs, the timing of the source data is not quite as critical for the operator effort as for the maintenance analysis. The operator training requirements are derived from the mission analyses and the maintenance training requirements are identified as part of the logistics support analyses. The mission decomposition and operator task timeline analyses forms the bulk of the system level operator tasks. The final recommended media functional and fidelity requirements eventually require detailed weapon system design data. However, in the early stages of the program, many of the operator tasks can be identified by parametric analysis and by reviewing the operational requirements. On the other hand, the Logistics Support Analysis (LSA) provides the source data for the system level maintenance tasks. In addition,

Table 2, Media Selection Models

Media Selection Model	Type of Behavioral Statement	Learning Classification Required	Phases of Learning Required	Type of Decision-Making Required
3306 TES	Skills & Knowledge	No	No	Flowchart
AFM 50-2	Learning Objectives	Yes	Yes	Multiple Form
AFP 5-58	Learning Objectives	Yes	Yes	Worksheet and Table
Anderson	Learning Objectives	Yes	No	Flowchart
AIMS	Tasks or Learning Objectives	Yes	No	Matrix with Weights
TASCS	Learning Objectives	No	No	Matrix with Weights

technical publications are not considered a prime source for maintenance tasks during the early stages of the EMD phase, since most of the weapon system's technical publications use the same LSA as a source. Most ISD applications have been based upon the premise that all the tasks required to maintain and operate the weapon system are already known. This is not the case in concurrent development.

TRAINING MEDIA REQUIREMENTS

The development of training media receives considerable attention due to the perceived cost and high visibility, particularly in air crew training devices. The ISD process and resulting trade studies define the training requirements for use by device/media designers and builders. The new training system is no exception.

Media Functional Requirements Development

The ISD process identifies the functional requirements of the training media. Functional requirements encompass the identification of training requirements, constraints, normal and abnormal scenarios, weapon system mission and functions, functional characteristics, modes of operation and instructional features. The ISD analyst must provide requirements that reflect the state of the weapon system design to support concurrency goals.

Media Fidelity Requirements Development

The ISD analysts define the degree to which the training device or media represents the actual system. There are two aspects to hardware fidelity; functional fidelity and physical fidelity. Functional fidelity refers to how well the media must replicate the actions and/or responses of the weapon system system, subsystem, or component. Physical fidelity refers to how closely the training media must resemble the weapon system components in size, shape, and appearance. The difficult aspect of defining these attributes in the new training system comes in trying to define them against an evolving weapon system design.

Media Development Flow.

The functional and fidelity requirements are defined in the Training System Functional Requirements and Recommendations Report, which documents the training

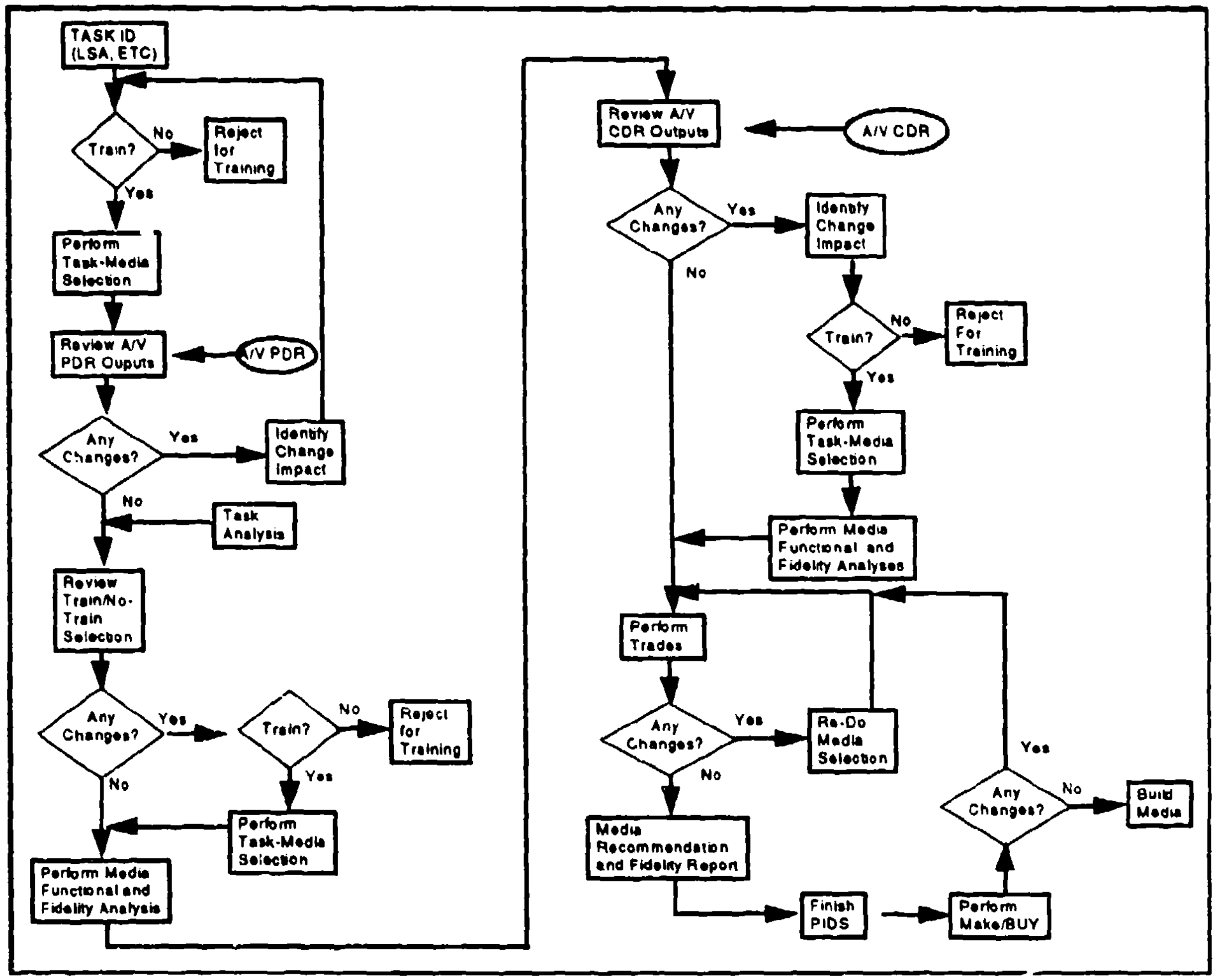


Figure 1 Correlation of ISD Processes to Weapon system Development

requirements to be used by the design engineers to develop the training media/device prime item development specifications. It also reflects the output of the ISD process as modified by the task-loading trade studies, the training effectiveness trades, and the cost-benefit trades. The media development progresses from preliminary requirements to a prime item development specification (PIDS) at the preliminary design review (PDR). The drawings are reviewed at the critical design review (CDR) and, after approval, the fabrication and/or procurement phase is initiated. Formative evaluation takes place as part of the integration and test phase. Summative evaluation typically cannot be completed until after the media is fielded. Part of the summative evaluation is to ascertain the currency with the fielded weapon system.

CORRELATION TO WEAPON SYSTEM DEVELOPMENT

The schedule for development of the new training system provides for incremental outputs to support decision-making points in the program. The new training system ISD process was designed to match the data to be available with the Weapon system Design Reviews (see Figure 1).

When the EMD phase started, we planned to provide a preliminary identification of the media to be used based on the outputs of the Weapon system PDR. Early identification of tasks and media was needed to support weapon system hardware and software reuse

definition. The quality of the output of the decision models would be directly related to the quality and quantity of the input data.

Weapon system Preliminary Design Review

The Weapon system Preliminary Design Review (PDR) is the basis for the first submission of the logistics support analysis data. The logistics analysts plan a two stage approach to developing the logistics support analysis data. The first, task identification, will be at PDR and the second, full task analysis, will be at the weapon system CDR. The logistics analysts, in their task identification phase, had only planned to develop task statements, logistics control numbers, and task codes as part of the task identification process. None of the available train/no-train or media selection models worked very well, if at all, with this limited amount of data. To make a train/no-train decision and select preliminary media with this limited amount of source data would require subjective decision-making on the part of the analysts. This was not desirable, particularly when a subcontractor was performing the maintenance front-end analysis for the training team.

Selection of the train/no-train and media selection models is intrinsically linked to the ISD tool or procedures selected for use on the program. The models have to be incorporated, or capable of being incorporated, in the tool procedures, or it was inappropriate for use. This is especially critical when you are working with analysts who are unfamiliar with the weapon system.

Weapon system Critical Design Review

The second LSA phase completes most of the remaining task analysis, including the identification of sub-tasks, sequential task descriptions, person identifiers, work area codes, number of persons per skill specialty, and performance standards. Anticipated schedules for completing the two stages were developed (see Figure 2). The first line represents the projected completion of the task identification [illegible]. The second line traces the projected task analysis completion activities.

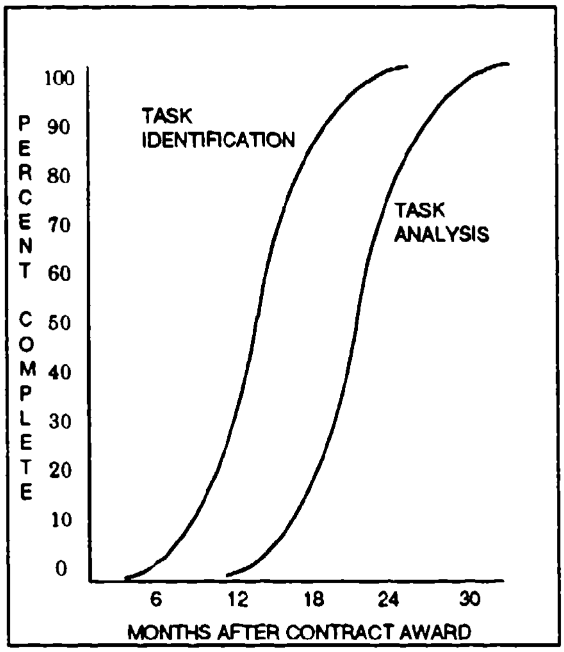


Figure 2 Projected LSA Schedules

Weapon system Deployment

During the deployment of the weapon system, the training system is required to maintain concurrency as the aircraft are deployed to operational units. As new operation units are activated with the delivery of new aircraft, the training system must expand to meet the increased support requirements. Considerable planning goes into the fielding of the training system. The turn-on of the device production lines to produce additional quantities is an example of the planning required. In fact, we found that we probably will have to start ordering long-lead items for additional training devices to support deployment even before the EMD phase has been completed.

System Support

Once the weapon and training systems are deployed to the field, they must be operated and supported. In the weapon system program, the training devices will be supported by contractor logistics support. The operation of the "schoolhouses" for

the operators and maintainers is still undecided. The ongoing changes to courseware and training media will be supported through operation of a Training System Support Center. Even at this phase, the ISD analysts will still be there, identifying changes needed to the training system in response to changes in the weapon system.

CONCURRENCY REQUIREMENTS

Engineering and Manufacturing Development Phase

Integrated Management Plans (IMPs) and Integrated Management Schedules (IMS) were used to manage the weapon system program during the EMD phase. These, in conjunction with the various specifications are used to control the development of the Weapon system, Support System, and Training System.

Adapting to Change - An essential element in the ability of the new training system to meet its goal of concurrency with the new weapon system is the use of integrated product development teams (IPDTs). The IPDTs allow the ISD analysts the opportunity to observe and participate in the evolution of the weapon system design. They are better positioned to support major changes in the overall program. A rephasing, conducted in late 1992 and early 1993, impacted the weapon system and Training Systems schedules. The weapon system CDR was extended by more than 12 months. The Training System R/DRU, Design Requirements Review (DRU), PDR, and CDR had to be extended accordingly (see Figure 3). This also resulted in extending the Training System PDR and CDR. The Training System is data-dependent upon the weapon system design and LSA and is, therefore schedule-dependent upon the weapon system.

Integrated Product Teams - Integrated product teams are used in the development of the weapon system and Training System. Concurrent development requires close coordination and careful planning by the IPDTs. The IPDT leader is a key factor in the success in meeting team goals and schedules. Experience in prior programs will greatly ease the difficulties IPDTs encounter in forming goals and establishing procedures. The leader becomes a defacto manager by accepting the responsibility for team performance. This includes the old management functions of planning, organizing, directing, and controlling, albeit modified by the need to participate in a teaming environment. This can be complicated when the mode of decision-making is by consensus.

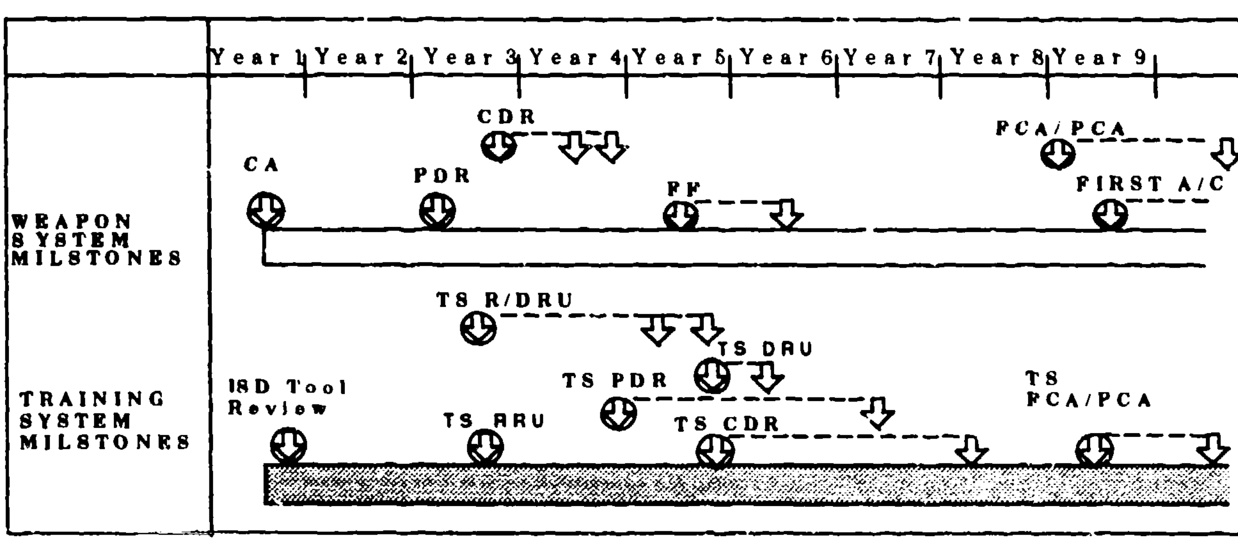


Figure 3. Weapon System Schedule Changes Impact Training System

Analysis and Integration - In the new training system, the responsibility for the ISD process was assigned to the Analysis and Integration (A&I) Team. The Training System A&I conceived and produced a Training System Development Plan (TSDP) that addresses the development of the training system from the beginning of the Engineering and Manufacturing Development (EMD) phase through the Operations and Support (O&S) of the fielded training system. The TSDP includes the ISD analysis, media trade studies, and other activities that are necessary, but not often clearly defined. The TSDP is closely coordinated with the contract statement of work and the IMPs.

Under the overall umbrella of the TSDP, a Training System ISD plan was developed that allocates the process into the first six of the TSDP's eight segments (see Figure 4.) In these segments, the ISD approach is modified to take advantage of the data available and still be concurrent with the weapon system scheduled event requirements. The ISD plan provides guidance and outlines the modified procedures used for operator and maintenance analysis. It further defines the data which will be provided to the media IPTs.

ISD Model Selection - The decision-support models selected for use in the new training program were the Difficulty, Importance, Difficulty (DIF), for the train/no-train analysis, and the Automated Instructional Media Selection (AIMS) for the initial media selection. These were selected primarily because it was believed there would be sufficient source data available to support their use. In the final analysis, it was the availability of source data that led to the selection of these particular models.

Source data availability and reliability are key issues, particularly in a developing program. The aforementioned rephase was driven in part by the need to better match the training system milestones with the availability of reliable weapon system and LSA data.

Segment I
- Analysis Procedures
- Target Population Analysis

Segment II
- Task Identification
- Train/No-Train Analysis
- Instructional Methods Analysis
- Prelminary Media Selection
- Instructional Setting Selection

Segment III
- Skill/Knowledge Analysis
- Conditions, Cues, & Standards
- Objective Development
- Media Functional & Physical Fidelity
- Cost/Trade Studies

Segment IV
- Curriculum Design
- Courseware Design
- PIDS Development

Segment V
- Courseware Development
- Media Development

Segment VI
- Course Evaluation and Validation
- TMS/TSSC Integration Verification
- Final Validation

Figure 4. Modified ISD Process

In the new weapon system program, source data content was negotiated between the Training System ISD analysts and the Support System logistics support analysts. Data elements, such as frequency, condition code, hazardous procedures code, means of detection, tools/support equipment requirements code, and criticality, were added to the task identification phase to support the preliminary ISD train/no-train and media selection analyses. This allowed initial training and media data at an earlier point in the overall schedule than would have otherwise been possible. This brought the ISD process in closer synchronization with the weapon system design.

Maintaining Concurrency - To maintain the training system concurrent with the weapon system is extremely difficult during the EMD phase. As discussed earlier, the detail of the LSA data increases as

the design progresses from PDR to CDR. This means the ISD analysts cannot complete their initial training requirements definition until after the weapon system CDR, reflected in the rephase activities. The time delay to complete the analysis puts the design of the training system behind the design of the weapon system. As pointed out earlier, this means the ISD analysts are constrained to work with the data that is available to begin building the audit trail.

A decision must be made to "draw a line in the sand" and "freeze" the design of the training system against some baseline. In the case of the new weapon system program, this will probably be the eighth EMD weapon system article. This allows media designs to be developed and subcontracts to be let. However, the training system still has a requirement to be concurrent with the delivered operational weapon systems. The solution lies in the concept of "first built, last delivered." Most of the devices built during EMD will be delivered "in place" at the contractor's facilities and will be used to support ongoing engineering activities. It is these ongoing media engineering activities and close proximity to the weapon system and avionics engineering functions that allow the training system developers to identify, track and make the updates to the training media to support concurrency requirements.

Production Phase.

In an ideal world, there will be no change between the weapon systems developed during EMD and those built for the production lots. However, experience on past programs indicates this has little or no chance of occurring. Therefore, the training system must be designed to support change, or as the Training System Product Manager, R.M. Foley, says, "Allow it to evolve gracefully." A Training System Support Center is planned to support the ongoing requirements of the training system. Designing to support change is the essential ingredient in maintaining concurrency with the weapon system. Supporting this is the continuing requirement for conduct of the ISD process throughout the production phase. The ISD analyst must continue to identify changes in design, mission, and procedures and determine the impacts on training media and courseware. To achieve the concurrency, the ISD analysts must be closely associated with the configuration control process for the weapon system. The analysts must be involved early to assure some hope of being able to identify the changes, determine the new or changed training requirements, provide updated functional and fidelity requirements, and support development of hardware, software, or courseware changes.

Operations and Support.

Concurrency is still a requirement, even at this phase, and is more of a challenge. Far too often, the need for the ISD process is ignored, or receives low priority for funding at this stage in the program life. However, the requirement to identify changes and their impacts still exists. The Training System Support Center and the ISD process will be around until the last article stops performing, still working to maintain concurrency between the training system and the weapon system.

CONCLUSIONS

Large scale programs that require concurrent development of training systems with the operational system are not a boon to the training analyst who is responsible for implementing the ISD process, or for the training system media developers who have to deliver media concurrent with the weapon system. Trying to conduct an ISD analysis in a concurrent development environment is a challenge and can be likened to "hitting a rapidly-moving target." The availability of data and the ability to respond quickly to change are the key elements of success in maintaining concurrency with an evolving weapon system.

There were and will be more complications. For example, in the new program, the planned levels of LSA data did not materialize when expected, reaching only 52 percent by weapon system PDR. This implies a very large amount of detailed data

will be available at weapon system CDR. Projected manning for the subcontractor performing the maintenance training analysis must be closely monitored and scheduled. For this reason, a "level-of-effort" contract was written and negotiated with the subcontractor.

The use of decision-aiding models adds consistency to the analysis process. There is no "right" way to identify and select models for use in the ISD process. However, the timing, quality, and quantity of the source data will narrow the models which can be used. Agreements can and must be negotiated to obtain source data to support the various stages of analyses, as was done in the new weapon system program. This is especially critical with LSA data. Each program has unique requirements which are used to develop criteria for selecting the models used in the ISD process. These criteria must be identified and the various models applied against them. The result is an ISD process tailored for the individual program.

The use of integrated product teams allows the ISD analysts to be involved in the weapon system design process. This enhances their ability to respond to changes. Effective product team leaders are essential in this arena.

Concurrent development is not a boon to the ISD process, but neither does it have to be a bane. Good planning and organization can greatly ease the pain. Integrated product teams staffed with qualified participants with an aggressive and experienced team leader are essential to concurrent development. Above all, be flexible and plan for change, for changes will occur when least expected.

REFERENCES

1. AFP 50-68 (DRAFT) Information for Designers of Instructional Systems, Volume 3 Application to Acquisition, 15 November 1992.

2. AFM 50-2 (DRAFT) Instructional Systems Development, 1 November 1992

3. Matthews, J.G. (1988) Computer-Assisted Media Analysis Program, Colorado Springs, Colorado.

4. Prothero, R.M. "Training Analysis - Panacea or Placebo," paper presented at Interservice/Industry Training Systems Conference, 2-5 December, 1991, Orlando, Florida.

5. Sorensen, H. Barbara. "An Assessment of Training and Education Media Selection Models," paper presented at Technology and Innovations in Training and Education conference, 2-5 December 1991, Orlando, Florida.

6. Sorensen, H. Barbara, Robinson, Gerald L., and Park, John S. "Supporting the ISD-LSA Interface with a CALS-compliant Data Interface," paper presented at Technology and Innovations in Training and Education conference, 2-5 December 1991, Orlando, Florida.

7. Survey of Training Development Software Tools, The CALS Human System Components Committee December 1991, Burlington, Massachusetts.

AUTOMATED INSTRUCTIONAL MEDIA ANALYSIS: LESSONS LEARNED AND RECOMMENDATIONS

Larry Clemons
JWK International Corporation
San Antonio, Texas

ABSTRACT

The Automated Instructional Media Selection (AIMS) model was used to allocate selected media to specific training objectives for the Air Force Primary Aircrew Training System (AFPATS) Ground Based Training System (GBTS). This paper discusses why the AIMS model was chosen over other media selection models, how it was used, what modifications were made, and what modifications are recommended for further use.

Choosing the best media selection model, from the more than 30 available, requires a careful matching between model capabilities and unique program requirements. Once selected, some modifications to meet specific program requirements may be necessary. For the AFPATS GBTS, the AIMS model offered the flexibility to add or delete as many as 30 candidate media and 192 instructional characteristics. The media weighting factors and the use of program-specific instructional characteristics used in the AFPATS program are discussed in this paper.

The AIMS model maximizes the use of pertinent information by automating the non-judgmental, data manipulation tasks of the media selection process. User-definable media pools and editing functions provide flexibility in adapting the model to specific problems and changing technologies. In addition, the user-definable aspect allows for inclusion of any instructional characteristic. The flexibility in defining the data manipulation can account for wide variations in the depth of front-end analysis to be accomplished.

Use of the AIMS model for the AFPATS GBTS allowed for assessing various instructional media for psycho-motor and cognitive skills. This was accomplished through program modifications that separated performance objectives from knowledge-based objectives. The flexibility in programming the reports' function was very useful in analyzing candidate media from different perspectives.

The paper presentation associated with this abstract is presented with a demonstration of the automated software used to employ the AIMS media matrix algorithms. The automated software and AIMS modeling are available through the government.

ABOUT THE AUTHOR

Larry Clemons is Manager of JWK International Corporation's branch office in San Antonio, Texas. He has twenty plus years experience in analysis, development and management of Instructional System Development (ISD) as used in government, business, industry and private sector. While on active duty in the Air Force he was a Command Pilot and Senior Space Operations Officer. Mr. Clemons developed curriculum and instructed in Undergraduate and Pilot Instructor Training. He has operational experience in pilot training for both fixed and rotary wing aircraft. In addition, he was the Pentagon Action Agent for ISD as used in the Air Force, Air National Guard, and Air Force Reserves. Mr. Clemons co-developed the current revision plan for Air Force ISD regulations and manuals. He was the Program Chairman for the '91 Technology and Innovations in Training and Education (TITE) Conference and Committee Chairman for the '92 Interservice and Industry Training Systems and Education Conference (I/ITSEC).

AUTOMATED INSTRUCTIONAL MEDIA ANALYSIS: LESSONS LEARNED AND RECOMMENDATIONS

Larry Clemons
JWK International Corporation
San Antonio, Texas

INTRODUCTION

The use of the AIMS model for media selection is better understood in the context of where and how the selection occurs in the instructional design process. Figure 1 shows the Air Force five-step Instructional System Development (ISD) model. Except for placement of a few steps in the process, this is very similar to a generic ISD model used by most businesses and universities.

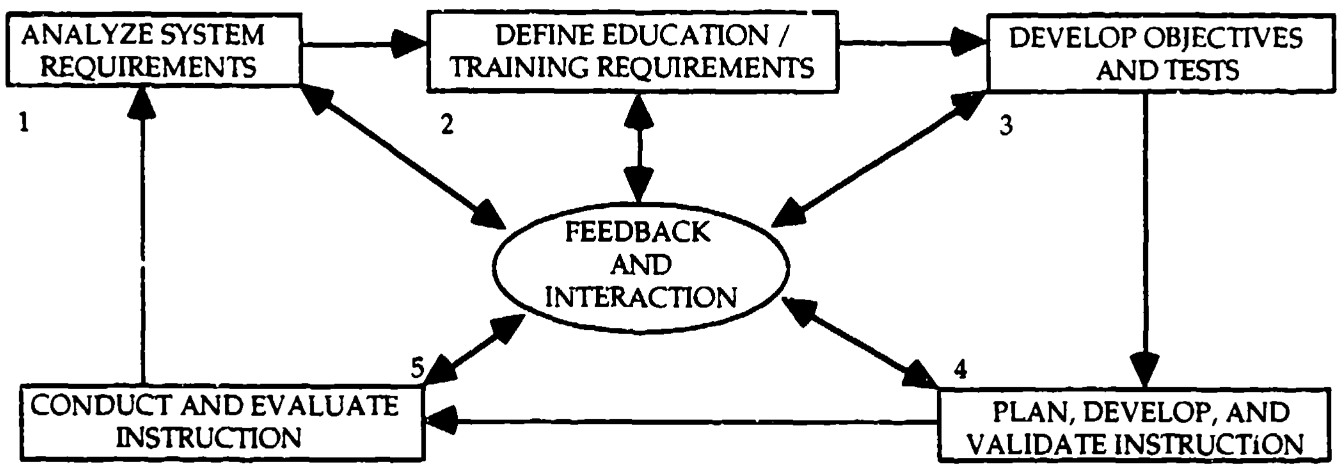


Figure 1. Air Force ISD Model

The information processed in the AIMS model is only as good as the quality of the data collected and analyzed in prio. steps.

The following is a brief description of the critical data required from the ISD prccess prior to entry into the AIMS model.

- Analyze System Requirements. A detailed listing of all psycho-motor, cognitive, and affective skills required to accomplish the task elements ot a job or mission.
- Define Education-Training Requirements. Translate task listing into Job Performance Requirements (JPRs) for those tasks selected for training. JPRs detail the specific behavior, the conditions under which the behavior is accomplished, and a standard which indicates how well the task/behavior is to be performed/accomplished. JPR support data include the specific skills, knowledges, and attitudes required to accomplish each task.
- Develop Objectives and Tests. Translate JPRs into objective statements, define the methods of instruction, and detail the evaluation methods. Support data for the objectives include a detailed listing of all instructional characteristics involved in the teaching of each objective.

This summary completes the first three steps of the five-step process and is sufficient data to enter instructional characteristics and objectives into the AIMS model. A parallel effort in technology assessment and media search is accomplished in order to arrive at a list of

candidate media to be loaded into the AIMS model.

MODEL SELECTION

More than 30 media selection models currently in use by the military services and non-military organizations were reviewed for use in the AFPATS program. Most of the models reviewed were based on either the original concepts or "spin-offs" of media selection processes proposed by Gagne, Reiser, Briggs, Rayner, Kemp, Anderson or Wager, all of whom are considered to be pioneers and innovators in the field of media selection modeling and applications. The purpose of these models is to provide a logical basis for matching training objectives with the most appropriate instructional media. During this review process, two facts became evident: 1) despite their fundamental similarities, most models are developed to satisfy the requirements of a particular training program, and 2) consequently, no two models are exactly alike. The AIMS model described in this paper is consistent with these findings. The basic framework of the AIMS model was applied to the specific needs and characteristics of the AFPATS Training Analysis database.

The model chosen for this project had to be:

- capable of handling the large media pool needed for a complete program of pilot training;
- a matrix model, allowing each objective attribute to be matched easily with the most appropriate medium;
- quantitative in nature, in order to generate interpretable ,rank-order recommendations for instructional media;
- automated and thereby capable of handling a large amount of data quickly and efficiently.

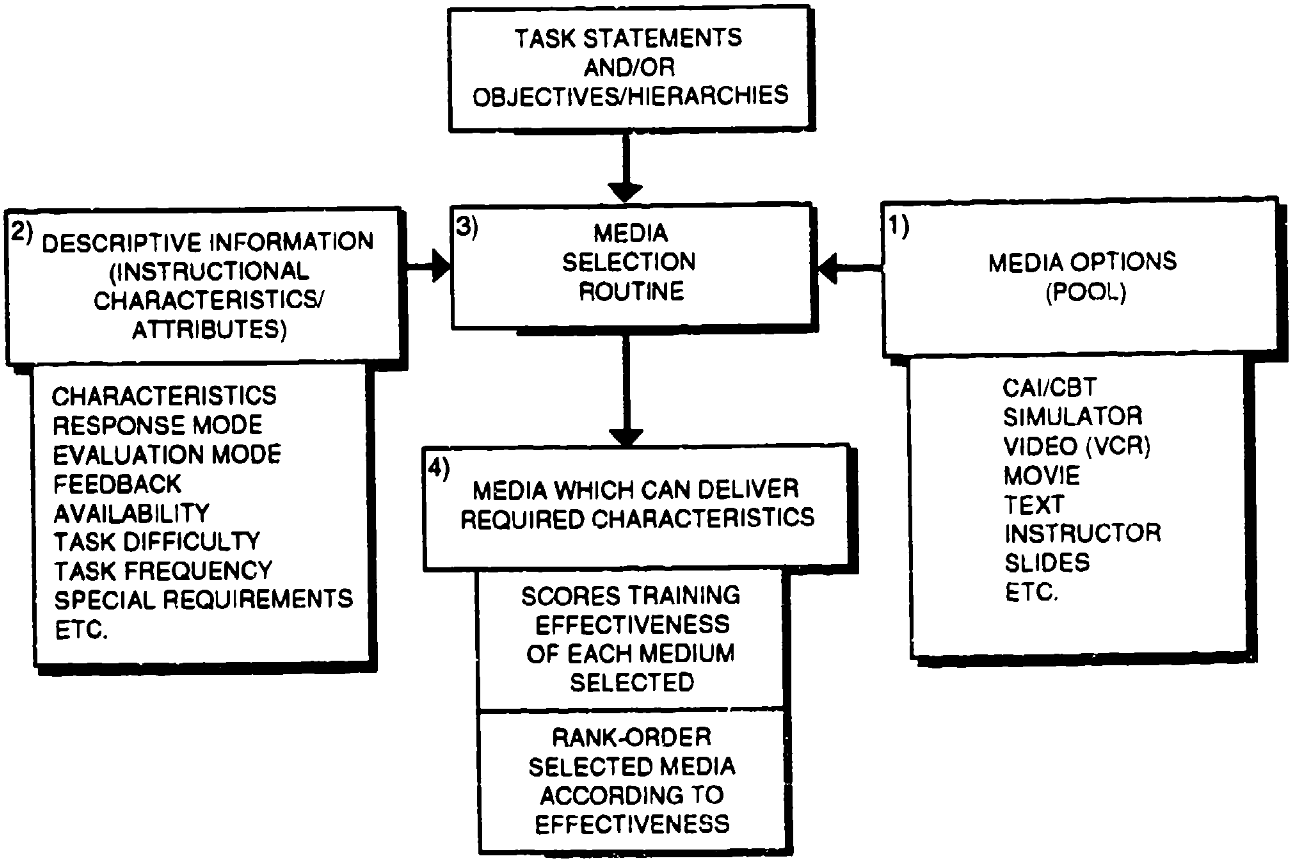


Figure 2. Automated Instructional Media Selection Model (AIMS).

The original AIMS model was developed by the Naval Training Equipment Center (renamed the Navy Training Support Center), Orlando, Florida, in 1983, for in-house use in developing curricular material for the various Navy training programs. The model has been revised and modified many times since 1983, but the basic premise and foundation of the model remain unchanged.

The AIMS model begins with the training analysts who, with input from subject matter experts (SMEs), determine three key components of the media selection process: 1) the set of attributes of all training objectives, 2) the best media pool for the given range of training needs, and 3) the extent to which each medium can address each attribute. Next, the automated part of the process allows the computer to integrate and tabulate this information, leading to a rank-ordered list of media recommended for each training objective. With these fundamental elements intact, the AIMS model was further developed and refined by training analysts and programmers at JWK for use in the Computerized Instructional System for Tasks, Objectives, Media, and Syllabi - Clipperized (CISTOMS-C) database system.

In Figure 2, the media selection routine (block 3) searches each objective (top block) for which instructional characteristics (block 2) are marked as required to teach that objective. The weight factors for the selected characteristics are then added to a total score for each candidate medium (block 1) in the media pool. The results (block 4) give total scores and rank order of recommended media for each objective.

MEDIA POOL

There were three primary data sources to determine which media would be included in the AFPATS media pool. The first source was information provided by the Technology Assessment performed for Training System Basis Analysis. The applicable areas of that report were extracted for use in the AFPATS media pool. The second source was a detailed review of the current training system achieved through interviews with personnel from the 338th Training Support Squadron at Randolph AFB, Texas. The final source of data input was technology reviews at the '91 Technology and Innovations in Training and Education (TITE) Conference and the '92 Interservice & Industry Training Systems Conference (I/ITSC).

The initial media pool included the types of media in use at Undergraduate Pilot Training (UPT), Pilot Upgrade Training (PUT), and Pilot Instructor Training (PIT) and potential media that may be selected for future use with the AFPATS aircraft. Data for the media pool are based on on-site visits to both UPT and PIT training bases, current related literature review, and information from subject matter experts (SMEs). One of the advantages of the AIMS model and the CISTOMS-C database is the ease and flexibility of adding and subtracting media to the media pool as revised data become available.

INSTRUCTIONAL CHARACTERISTICS/ATTRIBUTES

The ISD process yields very detailed task/behavior characteristics (attributes). These attributes are specific to each education or training program. In defining the appropriate training characteristics/attributes for use in the media selection process, the following points were considered:

- the relevance of the attribute to selected media in the media pool;
- the inclusion of attributes which identify the cognitive and psycho-motor skills required for task performance;
- the inclusion of attributes which distinguish between GBTS and performance objectives;
- the inclusion of attributes to address complex issues related to student pilot and instructor pilot training, such as situational awareness;
- the inclusion of attributes which address necessary crew/flight interaction and/or communication issues; and,
- the inclusion of attributes which consider the instructional method and method of evaluation so there is consistency in the media selected for a given objective.

The attributes chosen for inclusion in the media selection process are entered in a matrix with the media pool (see Figure 3). These attributes

were developed from standard instructional characteristics lists such as those found in the five step ISD process in Air Force Manual 50-2 (Instructional System Development). A subjective weight is assigned by SMEs to each medium attribute according to each medium's ability to consider the instructional attribute. The weights are assigned using a 0-5 scale where '0' indicates that the medium has no capacity for delivering this instructional attribute and '5' means that the medium is highly capable of handling a particular attribute. For example, an audio tape would likely receive a weight factor of '5' for the attribute of "aural cues", but should be assigned a weight factor of '0' for the attribute of "visual ground detail". The weight factors in this matrix are entered into CISTOMS-C, on a medium-by-medium basis, using the detailed task information taken from task analysis records.

INSTRUCTIONAL CHARACTERISTICS	Still Visual	Dynamic Visual	Wide Field of View	Vibration	Aural Cues	Forming Associations	Using Rules	Manual Dexterity	Affective Skills Training	etc.,	etc.,
MEDIA	1	2	3	4	5	6	7	8	9	10	11
1. Reference Text	2	0	0	0	0	5	1	0	1		
2. Lecturer/Facilitator	0	2	0	0	2	4	3	0	4		
3. Audio Tapes	0	0	0	0	5	3	0	0	2		
4. CAI (DMM)	4	3	4	0	3	4	4	0	3		
5. Aircraft	1	5	5	4	3	2	4	5	3		
etc.,											
etc.,											

Figure 3. Sample Media Matrix

In Figure 3, the vertical axis is a list of the candidate media collected from technology assessments and other sources. The top horizontal axis is a summary list of objective instructional characteristics. The grid contains the weight factors (described above) at the intersection of each medium with each instructional characteristic. Figure 3 is only a sample matrix. The AFPATS matrix contains 30 media and 80 instructional attributes. The process of weighting media against objectives is an automated process conducted within the CISTOMS-C media selection model. In this process, each objective is considered on an individualized basis. For a given objective, if an 'X' occurred in a database field identified as an attributed field, then the weightings for each of the media choices are calculated. Any attributes lacking an 'X' for a given objective are not included in the calculation process. At the end of this process, the user is provided with a listing of media, ranked from highest total score to lowest total score (the scores are provided for the purposes of showing the range

of numerical values). At this stage in the process, the instructional designer/training analysts can change the media decision generated by CISTOMS-C (i.e., disagree with the medium selected for a particular objective), and assign the desired medium in place of the medium assigned by CISTOMS-C.

MODEL MODIFICATIONS

The CISTOMS-C Media Allocation Report (MAR) ranks all media for all objectives. In the case of the AFPATS program, the objective data base was split between psycho-motor skills (Flight Training Objectives) and cognitive skills (Ground Based Training Objectives). Using the AIMS matrix as a guide, CISTOMS-C was modified to separately consider flight training and ground based training objectives. For reporting purposes, the data base was split between those media appropriate for ground based (academic) skills and motor (flying) skills.

Therefore, the media allocation report for flight training objectives reported only those media appropriate to flight training (e.g., aircraft, simulator, part-task-trainers, etc.,). Likewise, the media allocation report for ground based training reported only those media appropriate to cognitive skills training (e.g., lecturer, text, film, CAI, etc.,).

In addition, some modifications were made to the AIMS model because it is not an "intelligent" system. For example, some motor skills tasks have to be taught in simulation devices. Tasks like "perform ejection," "engage the barrier during landing," and "shut down the engine in flight" are not practical for training in the aircraft. The AIMS process of adding numbers to show hierarchy does not account for logical decisions. The AIMS model would tell you that the aircraft is the best place to teach ejection. However logical that may be, such a training strategy is not practical. Although not specifically done for the AFPATS GTBS, a modification to "help" AIMS make logical decisions would be to create a special instructional attribute (i.e. "safety" for which specific media (aircraft) would be coded with a '0'). The coding is such that if a medium shows a '0' for the attribute "safety," then that medium is excluded from consideration. Further, the ease of reporting recommended media by the AIMS model was facilitated by carefully listing the media in an order that permitted the split between psycho-motor and cognitive skills.

FUTURE MODIFICATIONS

The experience of the AFPATS Training System Requirements Analysis (TSRA) uncovered some additional modification requirements that would enhance the AIMS model for future front-end-analysis efforts. For example, splitting the media into psycho-motor and cognitive skills yielded more accurate and easy to read reports. The same concept could be used to include an option to split the instructional attributes on the AIMS matrix. The main purpose here would be to separate not only psycho-motor skills from cognitive skills, but to isolate affective skills as well. It could be argued that, if so many splits of the data base are required, separate AIMS matrices should be created. However, this would eliminate the reporting of all media against all objectives. Giving the analyst the option to split portions of the matrix offers more flexibility in generating reports.

In summary, the concept in the AIMS model is very useful in ranking candidate media for training psycho-motor skills. Modifications of the concept can be made for training cognitive skills. Some further development may be necessary for incorporating affective skills.

REFERENCES

Bateman, R.P. (1993). Training System Basis Analysis Report (TSBAR). (Final Report: Air Force Contract F336547-89C-2157). Dayton, OH: JWK International Corporation.

Clemons, L. (1992). Objectives and Media Analysis Report (OMAR). (Final Report: Air Force Contract F336547-89C-2157). Dayton, OH: JWK International Corporation.

Clemons, L. (1993). Syllabus of Instruction for Specialized Undergraduate Pilot Training. (Final Report: Air Force Contract F336547-39C-2157). Dayton, OH: JWK International Corporation.

Clemons, L. and Bills, C. (1991). "Revision of Air Force Instructional System Development." Proceedings of The Society for Applied Learning Technology Multimedia Conference. Orlando, Florida, 42-50.

Department of the Air Force. (1986). Instructional System Development. Air Force Manual 50-2. Washington, D.C.

Naval Training Equipment Center. (1983). Automated Instructional Media Selection (AIMS). (Technical Report: NAVTRAEQUIPCEN 79-C-0104-1). Orlando, Florida.

Tennyson, R.D. and Michaels, M. (1991). Foundations of Educational Technology: Past, Present, and Future. Englewood Cliffs, New Jersey. Educational Technology Publications.

LESSONS LEARNED FROM A CBT DEVELOPMENT TEAM

Maj Peter H. Lhotka, 1LT Todd W. Gorrell, MSgt Troy L. Rose,
SSgt Jeanine M. Butler, Sgt Sheri A. Semrau,
Mr Curt L. Taylor, and Mrs Terry L. Smith

Det 1, 4444 Operations Squadron
CBT Division, Interactive Courseware Branch
Luke AFB AZ 85309

ABSTRACT

The purpose of this paper is twofold: to document the lessons learned during development of actual Computer Based Training (CBT) and to provide practical recommendations on how to meet the "challenges" of producing quality CBT. Topics of discussion will include resource acquisition, project development procedures, and courseware implementation.

The mission of the Interactive Courseware branch includes producing quality interactive courseware (ICW) to train a variety of tasks for fighter aircraft operations. The branch has developed lessons in fighter aircraft operations, avionics integration, and precision guided munitions delivery.

ICW recently developed three CBT lessons for the F-16 and four CBT lessons for the A-10. Both projects involved major upgrades to include substantial hardware and software changes. This paper incorporates the lessons learned from these projects in the following areas:

1) Resource acquisition - personnel expertise, hardware/software requirements, support from upper level managers and subject matter experts (SME), and funding.

2) Project development procedures - team development, design/ programming standards, review/ validation process, and project management.

3) Courseware implementation - courseware distribution and follow-up evaluation.

The paper focuses on meeting the "challenges" encountered during CBT development. It emphasizes recommendations designed to assist other organizations in the pursuit of developing quality CBT.

ABOUT THE AUTHORS - Det 1, 4444 Ops Sq, CBT Division/ICW, Luke AFB Az

Maj Peter H. Lhotka is Chief Interactive Courseware (ICW) Branch responsible for development of CBT development projects. He has spent the last 13 years as an Education and Training Officer working as a program manager, instructional system designer, instructor, and CBT manager. He received a M.A. in Counseling from the University of North Dakota.

1LT Todd W. Gorrell is the ICW Program Manager. For the past three years, he has worked as an Education and Training Officer in charge of project management for a variety of CBT projects involving major weapon systems. He received a B.S. in Business Management from Park College.

MSgt Troy L. Rose is the Noncommissioned Officer in Charge of the Computer Based Training Division. He has 16 years experience in flight simulation performing maintenance and training for aircrew members. He has spent the last three years designing, developing, and evaluating CBT courseware.

SSgt Jeanine M. Butler was the chief programmer. For the past three years, she has been a

programmer designing and writing code for a variety of trainers and CBT courseware. She is now a civilian graduate student pursuing a M.A. in Counseling Psychology.

Sgt Sheri A. Semrau is an Instructional System Development (ISD) training technician. She has been in the Air Force for six years. She spent three years as an instructor teaching prenatal care, diabetes, and fat reduction. For the past three years, she has concentrated on design, development, and evaluation of CBT courseware.

Mr Curtis L. Taylor (GS-12) is the Senior Air Combat Command Instructional Systems Specialist in the ICW branch. He has 26 years experience as an instructional psychologist, including courseware design, CBT, and flight simulation. He has a B.S. and M.S. in Experimental Psychology (Concept Formation and Human Factors) from Kansas State University and is a doctoral candidate in Educational Technology at Arizona State University.

Mrs Terry L. Smith (GS-11) designs, develops, and evaluates CBT for Air Combat Command. She holds Master of Science and Bachelor of Science Degrees in Education from Eastern Illinois University and Eastern New Mexico University, respectively. Her background includes serving as an Air Force instructor, instructional system designer, and courseware development manager.

LESSONS LEARNED FROM A CBT DEVELOPMENT TEAM

Maj Peter H. Lhotka, 1LT Todd W. Gorrell, MSgt Troy L. Rose,
SSgt Jeanine M. Butler, Sgt Sheri A. Semrau,
Mr Curt L. Taylor, and Mrs Terry L. Smith

Det 1, 4444 Operations Squadron
CBT Division, Interactive Courseware Branch
Luke AFB AZ 85309

INTRODUCTION

The Interactive Courseware (ICW) branch at Luke AFB was created in 1986 to provide a single point of contact for Air Force Operations Training Development teams, courseware contractors, video/film production crews, and laser videodisc replication companies. ICW reduced duplication of effort and made the scheduling of flight crews, aircraft, simulators, and weapons load teams supporting courseware development more efficient. We supervised and monitored the development and final production of over 200 CBT/IVD lessons supporting flying training for A-10, F-15A, F-15E, F-16, and Low Altitude Navigation Targeting Infrared Night (LANTIRN).

Our mission has expanded from providing design guidance and quality assurance for F-15 and F-16 CBT contractors to in-house research and development of one-of-a-kind CBT for Air Combat Command (ACC). Projects have increased in complexity starting with basic introductory lessons and continuing up through lessons providing emulation of control display simulations.

The ICW mission consists of the three areas shown in figure 1. Forty percent of the work concentrates on providing consultation and training. The remaining sixty percent is divided between maintaining an in-house production capability and performing research and development on emerging industrial technology to support quality improvement.

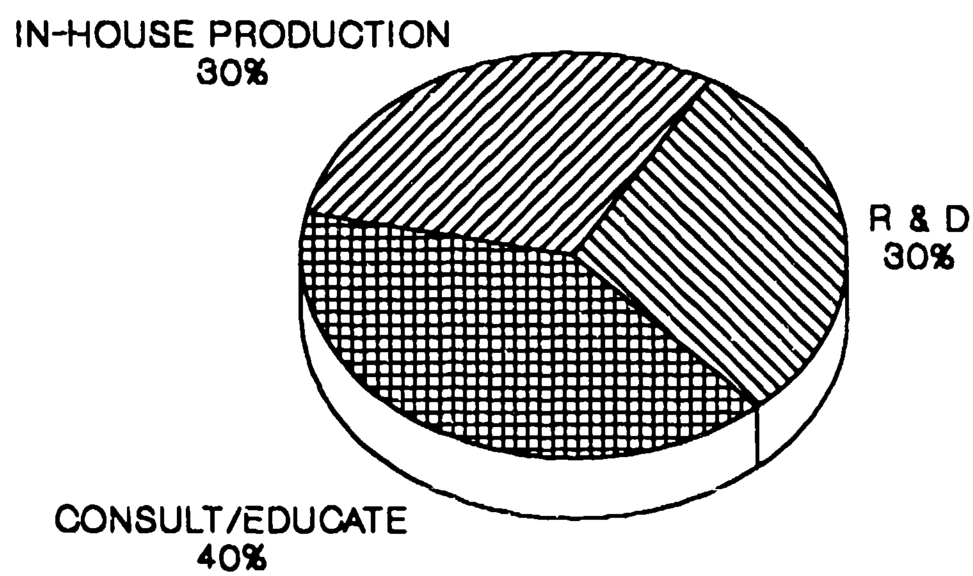


FIGURE 1.

This three-pronged approach has ensured that ACC combat aircrew Operations Training Development (OTD) teams get the benefit of our experience. Our goal is to be in a position to provide guidance and assistance at any level necessary. We do this in three ways. First, we provide consultation and training on ISD issues such as team development, task analysis, media selection, and instructional strategies. Next, we conduct in-house production to fill the contract gap. We respond to the needs of producing one-of-a-kind, short suspense courseware of the type that a contract can not cover in a timely manner. Finally, through formal research and development, we maintain proficiency with a rapidly changing medium and make that knowledge available for all OTD teams.

During the transition from courseware review to bonafide CBT development, we have supported specific training requirements identified by ACC training units. The CBT courseware, mainly for emerging weapon systems, has ranged from basic to complex skills required for fighter aircraft operations. An important part of this process has been the operation of an aggressive critique program. To improve the CBT developed in-house, we have utilized feedback from the student critique program.

A number of lessons were learned from our experience of building a development capability from the ground up. These lessons learned have greatly increased our capabilities and are the focus of this paper--to provide lessons learned in the form of recommendations to assist other emerging or active CBT development teams. This paper will specifically address CBT development capabilities in the areas of Resource Acquisition, Project Development Procedures, and Courseware Implementation.

RESOURCE ACQUISITION

Prevent Hardware from Driving Design of Courseware

In addition to human resources, we deal with hardware and software resources on a continuing basis. When selecting these resources, take into account the needs of the users and developers alike. We are often confronted with difficult decisions related to the hardware in current use. Do we design courseware according to available hardware in the field? This question is important because of the incredible volatility of computer hardware and software. Often, we have to compromise or make trade-offs to meet constraints or to conform to baseline hardware requirements.

When designing courseware, remember the importance of meeting the training requirements. The goal is to get the best training at the least cost. This often requires compromise because the best authoring program may not be compatible with the hardware in the field. We learned from experience that these issues need to be resolved early in the process. In the Air Force, we are faced with a limited budget and baseline hardware constraints. This situation dictates an approach using careful up-front planning to ensure the courseware design conforms to available hardware and effectively meets the training requirements. On a recent project we developed lessons that some field units could not run on their hardware. If a hardware upgrade is required to meet training requirements, one remedy is to budget for dedicated CBT machines early in the process or coordinate specific hardware requirements with the users. Dedicated CBT machines meet hardware requirements for running lessons, enhance the training environment, and provide greater accessibility for students to use the courseware.

Hardware upgrades are often required to take advantage of evolving technology and to raise the quality of the CBT package. When faced with hardware obsolescence, CBT developers and users alike must keep pace. For example, CGA was wonderful in it's day, but, a bit ineffective for multimedia presentations. It is critical to fully analyze the requirements early in the process and consider all relevant options before locking into a decision. For example, Interactive Video Disks (IVD) are effective for static lessons but not for courseware that changes frequently. For lessons that change frequently, photographic quality graphics have been less expensive and easier to change in our development process; however, a move from EGA to VGA is required. The key is to fully analyze the requirements before deciding on hardware/software needs.

Analyze Requirements Before Selecting Authoring Software

When selecting software, take steps to ensure the software meets specific criteria.

1) Analyze the training/design requirements to make sure the authoring program conforms to the requirements. Determine early what you want the authoring package to do. For example, if tracking student data is not important, why buy a package that is Computer Managed Instruction (CMI) intensive? There is a tendency to make snap decisions based on impressive technology rather than what specifically meets the needs.

2) Look for a vender who has a track record for providing needed user support. Nothing is more frustrating than venders who promise a lot, but do not deliver.

3) Select software that does not have an excessive learning curve. Ensure the software has logical and well-documented commands with an effective tutorial. The best authoring package is often the one you are already familiar with if it meets lesson/design requirements.

4) Conduct an aggressive R&D program on a variety of authoring packages. R&D in this area will facilitate decisions on choosing an authoring package. The process of matching requirements to the best authoring package is simplified through smart R&D.

5) Preferably, choose software that has a diverse user base. This allows one to network with others when seeking solutions. For example, some venders have User exchanges and newsletters to support their customers.

6) Find run-time software that does not require excessive memory. Developers generally have access to more powerful machines than users. Therefore, we recommend testing the product on equipment which matches the end users. Everything must be transportable!

7) Choose an authoring package capable of importing/exporting multiple graphic formats. This will save time and provide flexibility when building graphics into the lessons. The bottom line is that the CBT team must do their homework during up-front analysis to ensure all requirements are laid out and understood.

Acquire Support from Upper Level Managers

One of the most critical resources on a CBT project is support or advocacy from upper level managers. Without this support, a concept for the greatest CBT training package in the world could be developed, but it wouldn't be funded or produced. It's like any other product that you market. There must be a clearly defined training requirement and advocates at the decision-making level. The need must be communicated to upper level managers who will hopefully become advocates for the training. Then, funding and tasking can take place. This process will assure the needed funds are provided for manpower, software, and hardware. This is a mandatory prerequisite.

Identify Specific Requirements and Responsibilities

When starting a CBT project, it is essential to have and attend user planning meetings to identify specific support requirements and responsibilities. We learned this lesson on a recent project involving an emerging weapon system. We signed up to produce CBT lessons for the project without attending key planning meetings. In retrospect, we needed to attend earlier planning meetings to determine CBT project viability, training requirements, level of effort, and support agreements. In the future, these meetings will be a mandatory part of the analysis stage for any proposed CBT development project.

We became involved in this recent CBT project involving an emerging weapon system late in the process. This circumstance turned out to be a serious drawback. We were not able to establish a written agreement concerning responsibilities among the major players. A written agreement, called a Memorandum of Agreement (MOA), would have prevented problems related to project responsibilities during CBT development. We learned this lesson after establishing verbal agreements to receive technical support from SMEs, then finding it difficult to get the promised support.

A CBT project team must often depend on other organizations to obtain expertise and the latest technical data. We assumed that organizations with the subject matter expertise

would have the same enthusiasm for the project that we did. Unfortunately, this was not the case and we faced numerous difficulties in obtaining SME support. Without weapon system advocates at higher levels paving the way, it can be extremely difficult to access required information. We needed leverage to assign a full-time SME to the project.

Because technical expertise was often not available or incomplete, we had to acquire substantial expertise on our own. This resulted in time wasted during storyboarding, development, and validation because information had to be continually revalidated as it was incorporated into the course. We often had to depend on SMEs who had only partial knowledge of the system resulting in duplication of effort. Complete expertise on an emerging weapon system is always difficult to get, but we highly recommend that specific agreements (such as a MOA) be made early to ensure maximum SME availability.

Forecast Funding Requirements in Advance

Devote considerable effort to accurately predict funding requirements 9 to 12 months in advance. On a recent project, we were surprised by the complexity and the numerous avenues to explore in funding a project. We learned to be careful and to consider all requirements early in the process. In the military, you may only get one chance to ask for money and time spent up-front to analyze requirements will make the difference. Take into account required travel, manpower, new equipment, software, personnel training, and contingencies to ensure completion of a quality project on time.

When analyzing budget requirements, contact all major players involved to discover available "Pots" of money to support your project; the findings may surprise you. We discovered four to five months into a recent project that the Systems Program Office (SPO) for the aircraft we were developing training for had funds to support our CBT project. A SPO is in charge of the cradle-to-grave operations for a particular weapons system. On this project it was important to explore all "Pots" of money. For example, we could not find money specifically earmarked for training, but we did find a "pot" of money for "Technical Support". Because of the lack of SMEs and technical data, money for "Technical Support" turned out to be a critical part of developing the CBT. We learned the importance of being creative and exploring all avenues during the initial stages of a CBT project.

PROJECT DEVELOPMENT PROCEDURES

Assemble a CBT Development Team with Required Expertise

What expertise is required to make an effective CBT development team? We have established a team of members with a variety of experience and expertise. Our ICW team initially consisted of two ISD training technicians and one instructional designer. The team has now expanded to include two education and training officers, two programmers, and an additional instructional designer. Based on our development experience, we recommend the following composition/expertise for a CBT development team:

1) Instructional Designers with extensive ISD and CBT backgrounds
2) ISD Training Technicians with authoring language experience
3) Programmers with authoring language experience
4) Imaging experts with scanning, graphics, and authoring experience
5) Graphics artists - (if possible)
6) SMEs assigned locally - (if possible)

This team composition provides the capabilities to meet all aspects of a successful CBT project. On recent projects, a graphics artist could have enhanced our flexibility to use a variety of graphics. A graphics artist on staff would be ideal, but we get along without one based on acquired expertise and extensive use of scanning photographic images. In addition, a local SME should be available for all CBT projects involving an emerging weapon system. We could have saved time and money by minimizing the number of significant lesson changes required during the later stages. For example, we had extreme difficulty obtaining accurate technical information during the storyboarding phase of a recent project. Because of the SME deficiency, we wasted a lot of time during development and validation--making technical changes to the lesson. This inconvenience could have been prevented by ensuring proper SME support early in the project.

Conduct an On-going In-House Training Program and Attend Professional Conferences

Another important program required for a successful CBT team is an on-going training program. As a lesson learned, we established an aggressive training program during the development phase of two on-going CBT projects. The time was well spent and paid dividends in improving our capabilities and teamwork. We recommend weekly training sessions of at least an hour in duration. Pay special attention to identifying in-house deficiencies for future training to enhance the effectiveness of the training program.

Professional conferences are a "must" for members of any CBT team. These conferences keep personnel current on technological advances and also further research and development efforts. Given the volatility of computer technology, we have learned that attendance at these conferences help keep us a dynamic, growing CBT team. Conference attendance provides us with a non-stop supply of new ideas to improve our procedures. We have budgeted for every CBT team member to attend at least one professional conference this year. In addition, we accomplish periodic reviews of other software (R&D) and read professional journals/off-the-shelf magazines.

Find Out as much as You Can About Your Prospective Audience

Is your audience enamored with technology or are they distrusting of it? Our audience consists of Air Force Fighter Pilots. They ride technology every time they climb into a cockpit. They embrace rather than resist technology in their professional lives. This carries over into their personal lives, as well. Every squadron boasts a number of computer hackers; many pilots have state-of-the-art PCs in their homes. Because the use of technology has become second nature to them, they have high expectations of what it can do for them. This intimacy with technology became a problem. Student feedback began to indicate a level of frustration with the system due to the fact that the 286s that were used to deliver the CBT operated too slowly. Our solution was to gather together as many 386s as our budget would allow and to place them in the learning centers. Students were quick to show appreciation for the faster computers and chose to use them rather than the remaining 286s whenever possible.

What type of motivation must be built into the lesson? We had incorporated IVD into the lesson to provide motivational video and to provide audio narration. The use of IVD appeared to be a great idea, but proved not to be as successful as we expected. Our fighter pilot audience was unimpressed watching somebody else fly and the narration held the student to a fixed pace. What he could have read to himself in a half hour took one hour for someone else to narrate aloud. Students were quick to point out that these gimmicks detracted from the point trying to be made and lengthened the lesson; thereby wasting their time. We re-examined the use of video and determined that there was not enough pay back to warrant the initial expense of development. We began to give greater consideration to presenting the content in a straight forward matter and quickly!

Establish Project Design Conventions Early

How is your finished courseware going to look? Style, structure, color, text, visuals, and effective use of the media need to be addressed. A project team provides the best overall product, but disagreement among member's efforts is to be expected. If a design plan is not formulated early on, cohesiveness will not be present in the final product. Then someone will have to go back and re-accomplish work to provide the desired uniformity. It is better to plan for the design early and save duplication of effort.

Team meetings were convened to surface design issues and to establish design strategies that we could all live with. We discussed text and graphic applications over the course of three or four short meetings. The outcome was an oral agreement to what the text, content, legibility, tone, color, font, size, and placement would be; what the graphic placement and colors would be; and that there would be an information header and student interaction footer. Later, our agreement was documented and distributed to all team members. Index cards with text styles; programming cues; and placement data were placed near each development station for quick reference. A screen layout template that could be used repetitively was developed to provide uniformity

to the lesson. Disk copies were made of this file and distributed to team members.

Use the Quality Improvement (QI) Process to Promote Teamwork

As a CBT project team, we make many decisions concerning hardware, software, design conventions, programming conventions, and project management. During team meetings, we surface a variety of issues and establish strategies to produce the best CBT possible. Generation of ideas and consensus-building are key aspects of our meetings. QI utilizes the ideas and energy from all members of a team to brainstorm, evaluate possible solutions, reach consensus, and solve problems.

We have found that the QI process is an integral part of operating a successful CBT development team. Especially from the standpoint that QI encourages creative ideas, keeps all members involved in the decision-making process, and works well for instructional design projects. There are many examples of how the QI process helps to promote teamwork; an entire paper could be devoted to this topic.

It is interesting to note that the steps in the ISD process match the steps of the continuous improvement cycle associated with the QI process. ISD is a total quality process. The update of the US. Air Force ISD process reflects the total quality approach.[1] We highly recommend the revised ISD process described in AFP 50-68. The revised AFP 50-68 includes specific volumes with guidance on CBT selection and ICW development/ management.

Decide on a Plan for Lesson Delivery

What equipment is available for lesson delivery? How much technical support is in place for setup and maintenance? Ask up front, before beginning the project, what equipment will be used to run the courseware so that you can identify your hardware/software baseline. For example, on one project we found the lesson had to run on 286s equipped with EGA monitors. The courseware would be used in a learning center with a dedicated manager and a maintenance support system already in place. The courseware would also be delivered to operational units to load to personal computers. Our baseline was established by the EGA monitors. They limited us to using only EGA images eliminating full color photographic images. Operational units would need courseware that was self-installing and problem free due to the lack of a dedicated computer manager. One important comment to add is once you have decided, stick to your original plan. If this is not possible, or if you have very good reasons for not sticking to the plan, expect cost overruns.

Ensure a Programmer is Involved from the Start

Computer programmers should be involved in every project as early as possible. We learned this lesson on a recent project; when two programmers joined the team halfway into a project, they pointed out several innovative ways to improve the lessons. The development process would have been more efficient if those ideas had been part of the original design. The programmer is in a good position to make decisions such as determining the functional level of the courseware, the amount of programmer analysis required, and the best estimate of project completion time.

Many project requirements using sophisticated authoring systems available today can be accomplished with a programmer acting as a consultant only. On the other hand, other project requirements mandate that the programmer perform the majority of development. The critical lesson learned is that courseware can be developed faster and more efficiently when programmers are utilized more effectively. The result is more time for researching ideas for other projects. During initial stages, programmers can determine the feasibility and time requirements of specific tasks. After project completion, programmers can be consulted for estimated completion times for language upgrades, equipment modifications/ upgrades, and errata changes.

Adhere Strictly to a Regular Data Backup System

One of our most important lessons learned is the importance of the creation and strict adherence to a regular data backup system. The more individuals contributing to the project (programming and authoring), the more chance there is of code overwriting code. Our last project involved five development stations and six individuals. Prior to adopting the following

backup system, we made several time-consuming errors. The system that evolved which proved to be the most successful was as follows:

1) One computer was selected as the master computer. When changes were made on the other computers, they were copied to backup diskettes and dated. Each individual made a personal backup disk as further precaution against losing data.

2) A programmer was selected to copy backup diskettes to the master computer at least once a week. Upon completing this, the entire directory on the master computer was copied to another diskette and dated. This individual would then take the backup diskette and copy it on to the other computers being used as development stations.

3) A project status board was kept in a central location with a list of the lessons in the course and a space for names of the individual working on that area. We found that when it was time to send courseware out for review or to be released, a complete review of the entire course each time should be accomplished by an objective party in order to catch cosmetic errors and/or branching problems. Consistent adherence to this step will save a lot of time and embarrassment later.

Conduct Thorough Documentation of Courseware

To ease the transition of new personnel, we found that thorough documentation of the courseware needs to be accomplished on an on-going basis. First, each individual program file should have a standard documentation heading that includes items such as purpose of program, program name, programmer name, date, compiler used, which module calls this program, and any other data deemed necessary. Comments should be included in the source code, so that an individual unfamiliar with the language can follow the basic logic. In addition, a flowchart which shows the general structure of the course is helpful, especially to new programmers who want to see an overall picture of what the courseware does. This can be accomplished either manually with a template or with one of the utility packages on the market designed to produce flowcharts.

Implement a Precise System for Courseware Review

After a lesson has been developed, it is then given to our Quality Assurance Evaluator to be placed in the review cycle. Each lesson is logged into a lesson tracking database maintained by ICW. We have learned that precise tracking is necessary to ensure lessons flow through the cycle in a timely manner. From there an Instructional Systems Specialist does an ISD and functional review (functional reviews are not done on Lesson Specifications and Storyboards.) The review process begins at the Lesson Specification stage and continues through out the lesson's life. We have found that if any step is skipped, the quality of the lesson drops significantly.

An ISD review is essential to ensure the objectives are clear and precise. The lesson content must meet the intent of the objectives. Does the lesson content relate to what the student is supposed to perform/know in the objectives? Objectives are developed through careful analysis of instructional requirements. We have found some lessons contain information and gee wiz stuff that does not relate to the objectives. As a rule, objectives should determine appropriate lesson content.

Functional reviews, including IVD, are also conducted on all lessons during the review cycle. The quality of video and audio is checked for several specific effects. We ensure the video is clear, in focus, and relevant to the lesson topic being discussed. We verify that the audio is grammatically sound and narration is clear with correct pronunciation. We also ensure the graphics being displayed relate to what is being discussed. Some of the lessons have contained great graphics (simply because someone liked them) that added nothing educationally to the lesson. Finally, check for proper branching. Does the lesson advance/backup and branch to the proper frame? The best people to check branching are not the developers. They tend to be too close to the project, and don't provide the objectivity required for a good review. We have an ICW section, called Quality Assurance and Educational Standards, specifically established for this purpose.

COURSEWARE IMPLEMENTATION

Label and Package the Courseware in a Professional Manner

How can you improve user acceptance? Create a better first impression of your courseware. We package our lessons in the same manner as commercially produced software. Therefore, if the recipient has had any experience with software installation, he or she will be able to put our courseware on his or her system easily. Lessons are zipped along with batch files for installation and menus. Loading the disk is just a matter of selecting the proper drive and then typing INSTALL. Normally, only one disk is used for each lesson. The disks are labeled with the Lesson, Date, and our address and telephone number. The disks are distributed in an envelope which also contains the same information as the disk label. In addition, instructions for loading are printed on the envelope.

Market Your Product and Provide Support

What improvements would the customer like to see? Solicit customer level of satisfaction and suggestions for improvement. By keeping in tune with your audience and their needs, you are able to better serve them. We literally keep in touch. We capitalize on networking to personally hand deliver our lessons to the user.

The process does not end with delivery of the courseware. Follow-up is a must! Telephone interviews are conducted to ensure the disks have been received and to offer assistance if needed. In addition, evaluation forms are mailed to the customers three to four weeks after receipt of the disks. Evaluations determine if any problems were encountered while loading the courseware to the system and how well the courseware meets training requirements.

Be sure to provide technical support after lesson delivery for solution to hardware/software problems. Ensure the users have access to members of the development team to resolve courseware problems.

CONCLUSION

The lessons learned in this paper were documented to assist other emerging and active CBT development teams. As part of our mission of documenting these issues, we have found a definite pattern to our lessons learned. The criticality of conducting adequate **planning and up-front analysis** consistently rises to the forefront as a key theme. When applying these lessons learned, the identification of requirements **early** in the process is the most important aspect of managing and producing successful CBT. We stress this issue based on our own mistakes and the in-house initiatives we have used to remedy this problem. Another key theme related to early identification of requirements is the use of the QI process. The importance of using the QI process to surface ideas, reach consensus, and solve problems is extremely helpful in promoting teamwork and finding creative solutions. Overall, this paper will provide CBT development teams with practical advice on how to produce more effective CBT and avoid many pitfalls.

REFERENCES

1. Golas, K., Schriver, S., Bills, C., and Bowden, P. "Update of the US. Air Force Instructional Systems Development (ISD) Process", Proceedings of Interservice/ Industry Training Systems and Education Conference, Nov. 1992.

ESTIMATING TIME TO DEVELOP INTERACTIVE COURSEWARE IN THE 1990s

Katharine C. Golas, Ph.D.
Southwest Research Institute
San Antonio, Texas

ABSTRACT

In this paper a methodology for estimating the time required to design, develop, and evaluate ICW products was prepared and evaluated by 20 ICW experts from industry and the government. The methodology will appear in the Air Force Handbook 36-2235, Information for Designers of Instructional Systems, Volume 5, Interactive Courseware Design, Development, and Management Guide. This handbook has been developed for individuals in the Air Force who are responsible for ICW efforts. Many of these individuals lack previous ICW experience. The methodology will be used as a guideline to help them in their efforts to estimate time to develop ICW.

This paper presents the original methodology as it was received by the expert reviewers, summarizes the reviewers' comments, and presents the final methodology as it will appear in AFH 36-2235.

ABOUT THE AUTHOR

Dr. Katharine C. Golas is manager of the Instructional Systems Section at Southwest Research Institute. She began her career in ISD in 1977, by using the Interservice Procedures for Instructional Systems Development Model to develop print-based exportable job training packages. During the past 16 years, she has directed over 7[illegible] ISD projects, including 20 interactive videodisc projects and ten DVI®* projects. She is currently directing research and development efforts using advanced multimedia training technologies. In 1992, she led a project team to redesign the Air Force ISD model and methodology. She has a Ph.D. and M.A. in Instructional Systems from Florida State University.

*®DVI is a registered trademark of Intel Corporation.

ESTIMATING TIME TO DEVELOP INTERACTIVE COURSEWARE IN THE 1990s

Katharine C. Golas, Ph.D.
Southwest Research Institute
San Antonio, Texas

INTRODUCTION

Many computer-based training or interactive courseware products have been developed since the mid-1970s. There is a great deal of data available regarding the time and costs required to develop those products. Some organizations have even developed automated cost-estimating tools to serve specific purposes. However, a literature review revealed the lack of a generic methodology for estimating the time to develop interactive courseware (ICW) products in the 1990s. Such a methodology was prepared for Air Force Handbook 36-2235, Information for Designers of Instructional Systems, Volume 5, Interactive Courseware Design, Development and Management Guide. This paper describes that methodology and how it was evaluated.

The methodology that appears below was reviewed by 20 ICW professionals from various industries and government agencies. Their average number of years of experience in developing computer-based training (CBT) materials was 18. Comments received from the reviewers were analyzed, and the original methodology was revised to reflect their input. The revised methodology, as it will appear in AFH 36-2235, is included in this paper.

ORIGINAL METHODOLOGY

The original methodology for estimating ICW design, development, and evaluation time was based upon degree of interactivity, use of media, type of instructional content, and presentation format. A matrix was developed to serve as a baseline from which an organization could begin the process of estimating the number of hours it would take to develop one hour of ICW. The baseline estimate was developed for a "best case" scenario. A list of factors was then provided which affect the amount of time it takes to develop ICW. The idea of the methodology is to provide an organization with a starting point from which to tailor their estimate based on their personal situation.

Levels of Presentation

Three levels of ICW presentation were used to describe the degree of interactivity and use of media. The descriptions of the three levels given below were drawn from MIL-HDBK-284-1, ICW for Military Training, Parts 1-3.

Level I is the basic presentation. It is the lowest level of ICW development in that Level I lessons are linear (one idea after another) and are used primarily for introducing an idea or concept. There is little interaction other than the student touching the screen or using a keystroke or mouse click to continue. The media used are primarily graphics (not complex) and text.

Level II is the medium simulation presentation. This level allows the student to have increased control over lesson presentation. There is more interaction, such as using a light pen to rotate a switch. Computer-managed instruction (CMI) is implemented to keep track of and analyze student performance. The media used are audio, video, graphics, animation and text.

Level III is the high simulation presentation. This level provides a high degree of interactivity, extensive branching, maximum remediation opportunity (supports multiple levels of errors), real-time event simulation with minor equipment limitations, capability to interface with other output devices, and thorough CMI capability. The media used are full-motion video, audio, complex animations and graphics.

Types of Training

The learned capability—knowledge, skill, and attitude—was described in terms of three types of training objectives:

A **knowledge** objective involves the use of mental processes which enable a person to recall facts, identify concepts, and apply rules or principles. An example of a knowledge objective is knowing how fuel flows through an aircraft system. A person manifests knowledge through performing associated overt activities. Although knowledge is not directly observable, it is measurable.

Skill objectives are commonly described in terms of hard skills and soft skills. Hard skills involve physical or manipulative activities, such as operating a piece of equipment. Soft skills often require interpersonal activities such as conducting an interview or making a sales call. Both hard and soft skills are directly observable and measurable.

An **attitude** is a persisting state that influences or modifies an individual's choices or decisions to act under certain circumstances. An attitude objective represents a tendency on the part of the learner to respond in a particular way. An example of an attitude objective is choosing to wear a seat belt. Attitude objectives may be difficult to observe and measure.

ICW Format

Two ICW formats were described: analog and digital. With analog ICW, interactive videodiscs (IVD) are often the storage format for multimedia information. With digital ICW, full-screen, full-motion video, audio, still images, graphics and text are stored as digital data. Digital Video Interactive (DVI)® was described as the most common technology for developing all-digital multimedia presentations.

Factors That Define Best-Case Situation

Eight factors were used to describe the best-case ICW situation.

1. The ICW developer has in-house subject matter experts (SMEs).
2. The content domain is stable (i.e., the "system" exists and is not emerging).
3. The training content is well documented (task analysis completed, good technical materials).
4. The total ICW course length is 30-40 hours.
5. The ICW developer is familiar with the selected ICW software program and target audience.
6. A training needs assessment was performed, giving the ICW developer a good idea of the performance expected.
7. There is no requirement to develop to a MIL-STD such as 2167A.
8. The project team consists of individuals experienced with ICW design and production.

Estimate of Hours Given Best-Case Situation

Table 1 shows the estimated hours to develop one hour of ICW given the best-case situation, including the hours estimated for each level of presentation, type of training and format. These estimated hours were to be used as a starting point in estimating the time to develop one hour of ICW.

Factors That Affect Time Estimates

Ten factors were listed as affecting ICW time estimates. A table showed how many hours to add to the baseline estimate if the factors were not favorable. Estimates were also provided regarding the amount of risk on a scale of 1 (no risk) to 5 (high risk) associated with each variable. The ten factors and time and risk estimates are shown in Table 2.

EVALUATION STUDY

The original methodology was reviewed by 20 ICW experts with an average of 18 years of experience in designing and developing CBT. The reviewers and their organizations are listed below.

Reviewers

Ms. Virginia Anderson, Design for Learning
Dr. Alfred Bork, University of California
Dr. Frank Capuzzi, LORAL
Mrs. Lori Evans, US Army (TRADOC)
Dr. Peter Fairweather, Jostons Learning/WICAT
Dr. Dexter Fletcher, Institute for Defense Analysis
Dr. Andy Gibbons, Utah State University
Mr. Robert Huffman, Advance Development Group
Mr. Jay Jared, Boeing Defense and Space Group
Mr. Hank Kehlbeck, Booz, Allen & Hamilton, Inc.
Mr. Dan Kelley, Booz, Allen & Hamilton, Inc.
Dr. Dewey Kribbs, Instructional Science and Technology
Mr. Pete Larsen, Southwest Research Institute
Mr. Rod Lester, McDonnell-Douglas
Dr. Fred O'Neal, Jostons Learning/WICAT
Mr. John Payne, CAE Link
Dr. Larryetta Schall, US Army (TRADOC)
Dr. Sylvia Sharp, Consultant
Mr. Kent Thomas, Allen Communications
Dr. Lois Wilson, LORAL

Table 1. Estimate of Hours Needed to Develop One Hour of ICW (Best Case Situation)

Best-Case Estimate	Level of Presentation	ICW Format	Type of Training (Hours)		
			Knowledge	Skill	Attitude
50-200	I Basic	Analog	100	150	200
		Digital	50	100	150
150-400	II Medium	Analog	300	350	400
		Digital	150	200	250
200-600	III High	Analog	500	550	600
		Digital	200	250	300

Table 2. Factors Affecting Time Estimates in Table 1

Variables	Increase Hours By	Risk (Scale 1-5) None 1 2 3 4 5 High
1. Developer not familiar with target audience	5:1	1.5
2. Expected performance not known	45:1	5
3. Equipment is emerging	50:1	4
4. Poor documentation	20:1	3
5. No subject matter expertise	75:1	3
6. Must develop to a MIL-STD spec and deliver large amount of documentation	100:1	1
7. Developer not familiar with ICW software	100:1	2
8. Inexperienced project team	100:1	5
9. Video production contracted out	25:1	1
10. Programming contracted out	75:1	2

Summary of Reviewers' Comments

Levels of Presentation. Only three of the 20 reviewers commented on the description of levels of presentation.

- One said that most ICW is Level II.
- One said the levels of presentation were helpful and well written.
- One said it is difficult to apply the levels to any one piece of courseware—that there is usually a range of techniques used in an ICW program.

Types of Training. Regarding the description of knowledge, skills, and attitudes, seven of the 20 reviewers had comments.

- Three reviewers argued strongly that separating types of training into three categories is wrong, that in most ICW courses there is a mix of knowledge, skill, and attitude objectives. These reviewers agreed that the dimensions of skill, knowledge, and attitude are not independent. One wrote, "There should be a corresponding non-linearity in the numbers where they do interact. The problem is, I don't know anyone who would be willing to risk predicting, on the record, how they would interact." He went on to say that you should not pigeonhole materials into one of three categories for estimation purposes. On this same issue one reviewer said, "Every instructional product is an attitude product by default, because every one conveys an image, a style, and projects a persona. Every attitude product has some degree of skill building."
- One reviewer felt that knowledge lessons take more time to develop than skill lessons because skill lessons often involve teaching procedures which can be easily broken down into components and organized into logical presentation sequences. Also, knowledge lessons may require a measurement of understanding that can only be ascertained by careful monitoring of student performance, and because of this it is possible that more complex branching and feedback will need to be designed, thereby increasing development time.
- One reviewer wondered if "soft skills" was included.
- One said it costs a lot more to teach principles than facts.
- From a costing standpoint, one reviewer found the attitude category to be dubious at best. He said that it is not necessarily more expensive to develop attitude/affective materials. "It is usually more costly to develop DECISION-MAKING training, because this type of training is generally scenario-based, more freewheeling, and far more difficult to evaluate mechanically."

ICW Format. The format issue was hotly debated. Only four reviewers had no comments on this issue.

- Seventy-five percent of the reviewers disagreed that it takes less time to develop ICW for digital formats for the following reasons:
  - The standards and equipment for digital technology are still emerging, making the technology less stable.
  - There is a lack of data supporting the reduction in hours.
  - Initially there will be an increase in time due to learning curves.
  - By increasing options, working in all-digital formats can introduce complexity and slow the development process down.
  - The time required to devise the instructional design would remain the same for either format.
  - While digital video opens up some interesting possibilities for efficient shooting, storage, and replay of video, use of digital video may affect the quality of the final products.
- Two reviewers said the design time is the same regardless of format but there are differences in cost due to a reduction in time required to produce and edit video if digital formats are used.
- One reviewer said that many ICW programs being developed today combine the two formats by storing analog video and digital audio on a videodisc and digital text and graphics on the computer hard disk. "The real issue is the use of analog video, usually stored on a videodisc, versus the use of digital video, stored on a hard disk or CD-ROM." He believed it was wrong to focus on DVI technology in the description of digital formats because products such as QuickTime and Video for Windows are overtaking DVI as a cost-effective approach.

Factors That Define Best-Case Situation. Ninety percent of the reviewers commented on the factors that defined best-case situation.

- Two reviewers suggested that the estimated course length be raised from 30-40 hours to 100 hours and that the development process be accomplished over a period of one year.
- On the item regarding requirements to document to a MIL-STD, several reviewers suggested adding that using "best commercial practices" is acceptable for software and video production.
- Two reviewers suggested the addition of another factor, namely, that the selected authoring system and ICW hardware is mature and stable (i.e., not a beta version).
- Three reviewers commented that an optimal situation for ICW devalopment prevails if a lesson format and design strategy are agreed upon "up front" and the development process is standardized. One of these said it really helps if the customer has reviewed a prototype and "bought into" the design approach.
- On the customer involvement issue, three reviewers said that the design and development time can be greatly reduced if:

  - The customer works closely with the design team.
  - The customer has objective acceptance criteria.
  - The customer does not keep changing the person who is responsible for approving the product.

- One reviewer mentioned that a key factor to best case ICW development is "having all resources in place before you start!"
- Two reviewers said it helps if all aspects of the job are accomplished by one organizational team with none of the work being subcontracted.

Estimate of Hours Given Best-Case Situation. Of the 20 reviewers, only six had no comment on the estimates shown in the matrix in Table 1.

- One reviewer said that the numbers in the analog row were all high and should be reduced uniformly by about 50 hours.
- One reviewer said that 500 hours was too high for analog, Level III, knowledge domain lessons unless the criterion is "total interactivity" such as employed by games or discovery learning techniques. He said if this is not the case the number should be lowered to 350.
- Three reviewers said the numbers in the analog row were low and should be raised.
- Two reviewers wanted to know if evaluation was included as a part of the development process.
- Half of the reviewers disagreed that it takes roughly half as much time to develop ICW using all-digital technologies. Only one reviewer agreed with this estimate.
- One reviewer said that a training organization quoted a 1400:1 ratio for developing highly interactive DVI courseware.

Factors That Affect Time Estimates. As with the factors describing the best-case situation, approximately 90 percent of the reviewers had something to say about the factors that affect the time estimates.

- One reviewer felt that the hours should be increased by 10 rather than 5 if the ICW developer is not familiar with the target audience.
- Numerous reviewers commented that under no circumstances should ICW development be attempted if the expected performance is not known.
- Three reviewers were confused by the factor "equipment is emerging"; they didn't know if this meant the ICW development and delivery equipment or the equipment for which the course was being developed.
- Four reviewers said it is very difficult, expensive, and risky to try to develop ICW when the system for which the training is being developed does not exist. One reviewer said, "I would not do a fixed price contract if the equipment is emerging."
- Two reviewers felt that the factor "poor documentation" was unclear. They asked if this meant the source documentation did not exist or was totally inadequate, or was subject to change with great variability between iterations.
- The factor "no subject matter expertise" drew many comments. Two reviewers said if there are no SMEs the job can't be done at all (they assumed the use of customer SMEs). One reviewer said that when equipment is emerging there often aren't any SMEs and that this factor is "a killer" and that it should be rated more risky than all the other factors. Two

reviewers said that impact of lack of SMEs would depend on complexity of subject matter and adequacy of documentation.

- Regarding the factor of developing to a MIL-STD and delivering a large amount of documentation, three reviewers felt the number (increase by 100 hours) was too high. One reviewer said it was too high unless the developer had no prior experience with the MIL-STD.
- On the factor "developer not familiar with ICW software," four reviewers said that increasing the time by 100 hours was much too high. These individuals believed that given most authoring packages being used today, developers can be brought up to speed fairly quickly.
- On the factor "inexperienced project team," one reviewer commented that it depends on where the inexperience lies: "If it's design inexperience, the number is too low. If it's project management inexperience, the number is way too low. If it's authoring language inexperience, it's way too high."
- On the factors regarding contracting out video production and programming, half of the reviewers felt the estimates were too high, commenting that these factors do not impact development costs. One reviewer felt that contracting might even save money, depending on the vendor(s).
- One reviewer said the list of factors that affect time estimates should track with and match the list of best-case factors. This excellent suggestion was incorporated into the revised methodology.

REVISED METHODOLOGY

Based on the comments received from the panel of experts and follow-on discussions with the reviewers, numerous changes were made to the original methodology. The distinction between analog and digital was removed from the methodology due to the reviewers' concerns that the time savings have not been thoroughly demonstrated. The revised methodology is discussed below.

Table 3 provides a baseline estimate from which to begin the process of determining the total number of hours it will take to design, develop, and evaluate one hour of ICW. Program management time of approximately 10 percent is included in the baseline estimates. The hours provided in Table 3 assume a "best-case" situation as defined by the following factors:

1. The ICW developer is familiar with the subject matter, and has in-house subject matter experts.
2. The subject matter is not highly complex.
3. The instructional content is stable; that is, the system for which the training is being developed exists and is not emerging. Also, the tasks selected for ICW training do not continually change.
4. The instructional content is well documented. A training needs assessment and task and learning analysis have been completed, giving the designer a good idea of the performance expected and the tasks to be trained. The technical materials for the content are accurate.
5. The total ICW course length is 100 hours or more and the development process will be accomplished within one year.
6. The ICW developer is familiar with the selected ICW authoring software.
7. The ICW developer is familiar with the target audience.
8. There is no requirement to document to a MIL-STD such as 2167A, and best commercial practices are accepted for software development and video production.
9. The ICW project team consists of individuals who are experienced with ICW management, design, and development.
10. The selected ICW authoring system is mature and stable. No beta versions are used.
11. A lesson format and design strategy are agreed upon "up front," and the customer has "bought into" it. If possible, the customer has approved a prototype lesson. Also, the development process is standardized.
12. The customer works closely with the design team on a regular basis. The customer uses objective acceptance criteria and does not continually change the individual who is responsible for reviewing and approving the lessons.
13. All required resources are in place.

Table 4 shows how the time estimate per hour of courseware will increase if the 13 factors listed under best-case situation are not present. An example for using Tables 3 and 4 is provided below. Estimates are also provided regarding the amount of risk (on a scale of 1 to 5, with 1 being no risk and 5 being high risk) associated with each factor.

Table 3. Revised Estimate of Hours Needed to Develop One Hour of ICW (Best-Case Situation)

Best Case Estimate	Level of Presentation	Type of Training (Hours)		
		Knowledge	Skill	Attitude
30-200	I Basic	30	75	200
75-250	II Medium	75	125	250
200-600	III High	200	400	600

Table 4. Factors Affecting Time Estimates in Table 3

Factor	Increase Hrs By*	Risk
1. No "in-house" SMEs; must rely solely on use of customer SMEs.	25:1	2
2. Subject matter is highly complex.	100:1	4
3. Instructional content is unstable. System for which ICW being developed is emerging. Tasks for ICW constantly changing.	100:1	5
4. Inadequate documentation. No training needs assessment was performed. No task analysis or learning analysis data. Technical manuals/orders nonexistent or not helpful.	20:1	4
5. Total ICW course length <100 hours.	20:1	1
6. ICW developer not familiar with ICW software/authoring package.	15:1	2
7. ICW developer not familiar with target audience.	10:1	2
8. Best commercial practices not acceptable for video, graphics production, and software development. Must develop to a MIL-STD specification and deliver large amount of documentation.	50:1	2
9. Inexperienced project team:		
ICW designers inexperienced	80:1	3
ICW manager inexperienced	100:1	
ICW programmer inexperienced	60:1	4
10. Using a beta version of ICW software.	80:1	4
11. No prototype exists, no agreement "up front" on design strategy, no standardized development process followed.	50:1	5
12. Customer not using objective and consistent acceptance criteria. Customer unsure of what he wants and does not communicate with developer.	50:1	5
13. Required resources not in place at start of project.	20:1	1

*For each hour of courseware add the number of developmental hours shown.

Example: Assume that Level II ICW is being estimated to train a skill. The product to be developed is IVD, and the course length is estimated at 100 hours. You are familiar with the software and have experienced people. The programming and video production will be completed "in house." There are no MIL-STD requirements; however, no training needs assessment has been performed and the subject matter is highly complex (add 100 hours). You do not have in-house subject matter experts (add 25 hours). The instructional content is stable but the documentation is poor (add 20 hours). You are not familiar with the target audience (add 10 hours). Beginning with the number 125 (the hours it would take to develop one hour given the "best case" situation), you should add a total of 155 hours to the estimate, bringing the total up to 280 hours.

SUMMARY

Estimating the time required to design, develop, and evaluate ICW has been more of an art than a science. The intent of the methodology described in this paper is to make it more of a science. As the methodology shows, there are many factors to consider and the factors all affect or are affected by each other. For example, the ICW developer may have an excellent understanding of the subject matter because "in-house" SMEs are available, and yet may not have documentation in the form of task analysis data.

The methodology presented in this paper describes how certain variables affect a time estimate. The data collected from the reviewers indicated that most ICW programs consist of a variety of learning objectives, with a mix of knowledge, skills, and attitudes to be trained. Design strategies can even be mixed, where the lessons developed for knowledge objectives are linear with text and graphics and the lessons covering skills are more interactive and use audio and video.

The key factors that impact time estimation are complexity of the subject matter, experience of the project team, and stability of the course content. The ICW estimator should carefully examine the impact of these factors on each particular situation in order to arrive at a realistic time estimate. A realistic estimate is more likely to ensure development of a high-quality product, whereas a low estimate usually means that quality is compromised and customers are dissatisfied. One reviewer said it best:

> "Where ICW is concerned, I think that the focus should remain on the quality of the product and the effectiveness of the training, not on how fast you can do it."

APPRAISAL and TECHNIQUES for DIGITAL AUDIO in INTERACTIVE TRAINING

Lieutenant Commander John H Hall Royal Navy
Office of the Flag Officer Submarines
Gosport, U.K.
and
Dr. Ann E Barron
University of South Florida
Tampa, Florida

ABSTRACT

The increased availability of moderate cost, good quality, digital audio computer cards and peripherals has enabled trainers and instructional designers to realise the potential of random access audio for computer-based training (CBT) and other multimedia applications. There are, however, few guidelines for instructional designers to follow when incorporating audio into interactive lessons. This paper provides an overview of digital audio in interactive courseware. Hardware and software issues are analyzed, and the advantages and disadvantages of incorporating digital audio are outlined. In addition, areas of future research needs are investigated.

ABOUT THE AUTHORS

Dr. Ann Barron is an Assistant Professor in the Instructional Technology programme of the University of South Florida, Tampa, USA. She has been actively involved with the design and development of interactive courseware in military, commercial, and educational environments for the last seven years. Dr. Barron is a frequent speaker on new technologies at national, and international conferences. She has published numerous articles and recently published her first book, "New Technologies for Education: A Beginner's Guide".

Lieutenant Commander John Hall is the Computer Based Training and Simulation Officer to the Flag Officer Submarines. He is responsible for the procurement, development and management strategies of FOSM's CBT systems which are located at dispersed sites, and has been the project manager for a variety of ASW training systems over the last seven years. Recently loaned to the US Navy to provide a consultancy service for training methods which he had developed for the Royal Navy, Lt Cdr. Hall has had several papers published internationally on Naval training and interactive courseware. He has over 15 years experience of Naval technical training, including operational deployments, in addition to nine years in the electronics industry.

APPRAISAL and TECHNIQUES for DIGITAL AUDIO in INTERACTIVE TRAINING

Lieutenant Commander John H Hall Royal Navy
Office of the Flag Officer Submarines
Gosport, U.K.
and
Dr. Ann E Barron
University of South Florida
Tampa, Florida

INTRODUCTION

In the past, the best way to incorporate audio with a computer program was with analog audio on a videodisc. As a result, courseware design was constrained to a maximum of 60 minutes per disc side, if both audio tracks were fully utilized. In addition, the video and analog audio tracks had to be accessed together on a videodisc. This limitation caused many development hours to be devoted to disc geography in order to minimize search time and fill both audio tracks. Another problem [1] was that there was no audio available for most still frames, which often represented a large percentage of a lesson.

Several factors have recently made digital audio a feasible option for computer-based programs. First of all, the necessary hardware is now available at reasonable prices. Other factors contributing to the increased use of digital audio are the availability of large storage devices and the improved compression techniques for audio files.

ADVANTAGES OF DIGITAL AUDIO

A major advantage of digital audio in multimedia instruction is its flexibility. The recording and editing procedures are quick and easy. This allows courseware developers to update and/or replace files without mastering a videodisc. Authoring programs, such as Mandarin. TenCORE, or ToolBook, can easily and seamlessly call and play the audio files.

Another advantage of using digital audio in multimedia (as opposed to analog audio on a videodisc) is the fact that it can be recorded in-house. The ability to produce rapid prototypes where any voice is sufficient in order to check the flow of the courseware, is a major benefit. Once the audio content has been checked for accuracy, it can then be re-worked with a professional voice. For military applications, the use of digital audio makes revisions much easier, allowing for recording security classified narrations without the added expense and difficulty of pressing a classified video master. In addition, with digital storage, the amount of audio is no longer limited to 60 minutes per videodisc side - only to the amount of available memory on the host computer's hard disk.

DISADVANTAGES OF DIGITAL AUDIO

A major problem with incorporating digital audio with current conventional systems is that the storage of digitized sounds demands an enormous amount of disk space. This constraint requires developers to determine the optimal sampling rate based upon the courseware requirements and available disk storage. If the storage space is limited, the amount of audio incorporated may have to be decreased or the quality of the audio may have to be reduced.

Another disadvantage of digital audio in multimedia is the lack of standardization of hardware (especially for MS-DOS machines). Audio files which have been digitized with one type of audio board cannot generally be distributed for playback unless the same hardware is present in the delivery station. This restriction can impact the cost of a project and the range of possible delivery platforms, and may even lock the developer into using a particular product with associated reliance on a single source for technical support.

HARDWARE FOR DIGITIZING AUDIO

The process of digitizing audio with a computer requires an analog to digital (A/D) converter. To play back the digital files, the reverse process takes place and a digital to analog (D/A) converter is utilized. All Apple Macintosh computers contain built-in digital to analog converters to play audio files, and the new 'Macs' also provide analog to digital features to record files. Older Macintosh Computers can utilize a peripheral, at additional cost, such as a "MacRecorder" to digitize the audio and be compatible with more recent machines.

In the MS-DOS environment, many audio cards and peripherals are available at moderate cost, such as the SoundBlaster Pro, AudioPort, Turtle Beach, and IBM Audio Adapter (see Table 1). These cards and peripherals contain the converters needed to record and play the audio files. It is important to note that both the development stations and the delivery stations must contain the same audio devices as previously discussed. Many companies are developing external audio devices that connect to computers through a serial port in order to eliminate the present process required to install interior audio cards. This is seen as mainly benefiting the home and developer markets; the multi-station classroom in educational, commercial and military applications demands as much hardware integration as possible, especially where space is at a premium.

SOFTWARE FOR DIGITIZING AUDIO

Most digital audio devices provide software programs to control the recording and editing processes. These programs provide selections for sampling rates, volumes, and other recording variables. Preview options are available and in many cases, the developer can decide whether to record the files to RAM or directly to disk. If editing tools are available in the software program, digital audio files can be cut and pasted like text in a word processor. Many programmes allow the files to be viewed in a sound wave window at various zoom levels. Editing enables developers to rearrange words, delete excessive pauses, and compile new narrations without recording the files again. Time spent by an experienced editor can yield considerable memory savings which will produce benefits in reduced access time for file replay. However, there is a trade off between the time cost of the editor seeking to produce the 'perfect' presentation and the cost of memory saved.

AUDIO CARD	SUPPLIER	SAMPLING RATE (kHz)
DigiSpeech	DigiSpeech Inc.	1, 3, 4, 8
	Online Computer Systems	4, 6, 8, 12, 16, 18, 37.8
M-Audio	IBM	8, 11, 22, 44
MicroKey Audio Card	Video Associates	8, 16, 32
ProAudio Spectrum	Media Vision	6, 8, 11, 22, 32.44
SoundBlaster Pro	Creative Labs	4 - 44
SoundMaster II	Covox	4.6 - 25.4
VP625	Antex	8, 12, 16
MultiSound	Turtle Beach Systems	11, 22, 44.1

Table 1. A Selection of Audio Devices (USA) for MS-DOS Computers

SELECTING A SAMPLING RATE AND COMPRESSION TECHNIQUE

The sampling rate refers to the amount of digital information stored for each second of audio. For example, if a sampling rate of 4 kHz. is chosen, then information is recorded 4,000 times in each second. Thus the higher the sampling rate, the better the audio quality as more information is present. Sampling rates for a variety of commonly available audio cards are shown in Table 1. It is of note that most of the devices offer specific sampling rates, such as 8, 16, and 32 kHz. A few cards, such as the SoundBlaster Pro and Sound Master, enable the user to select a sampling rate anywhere within a defined range. There is, however, a trade-off: - high sampling rates require large storage space because there is more information to store and recall, therefore the sampling rate is often constrained to match the available storage. For example, only a 45 second file could be stored on 1 Mb of hard disk space if it was sampled at 22 kHz (Table 2). In general it has been found that 8-12 kHz sampling rates are sufficient for narrations; whereas music requires much higher rates (11-22 kHz) for any reasonable quality. This is derived from the general rule of selecting a sampling rate of twice the highest frequency required.

SAMPLING RATE (kHz)	STORAGE FOR 1 SECOND OF SOUND (kb)	SECONDS OF SOUND PER 1 Mb
22	22	45
11	11	90
7	7	135
5	5	180

Table 2. Sampling Rate and Storage

Another method used for reducing the storage requirements of audio files is to employ a compression technique. In some audio software programs, the developer may select a compression ratio, such as 4:1 or 3:1. In other cases, a built-in compression technique, called ADPCM (adaptive pulse code modulation) is used. Although compression techniques can yield significant memory savings, there is often an associated recall time delay as the audio files have to be decompressed or "unzipped". This presents real problems if the audio is required to be synchronised with the display of a graphic file, especially in interactive courseware when the selection of the graphic is under user control and cannot be predetermined.

DESIGN GUIDELINES FOR USING DIGITAL AUDIO IN MULTIMEDIA

A prime consideration when incorporating digital audio is to develop the applications on baseline hardware. In many cases, the audio files are loaded into RAM before they are played. If the development station has more RAM than the delivery station, some of the audio may be lost or the narrations may be choppy. Delays in unzipping compressed files can sometimes be reduced by using more RAM, but the advantages must be weighed against the extra costs involved; there will always be a point at which extra RAM yields insignificant gains due to other hardware imposed speed constraints. It is also best to avoid complex synchronizations (such as lip-synching) because it can be difficult to match the access time for the audio and video segments from different sources, and this again uses more memory.

Another recommendation is to keep the length of the audio segments short (5-7 seconds). This restriction will help to keep the files within the RAM limitations and also provide better chunking of the instruction. In addition, options should be available for the students to "repeat" the audio or "interrupt" it at any time. Some learners are very uncomfortable relying on audio inputs and may be more comfortable with visual cues and increased control of the audio.

Files generated by audio are proportional to the frequency range and dynamic range of the recording, and these factors must be constrained in order to keep files down to a manageable size. Most audio cards therefore limit the dynamic range to about 40 dB. When recording audio files, care must be taken to keep the amplitude of the digitized signal within the range of the audio card, otherwise distortion will occur.
The limitation of the frequency range is not as critical as the dynamic range. Even though the human ear can detect sounds ranging from 25 Hz to 16 kHz, most sounds necessary for a CBT programme fall within a much smaller range. For example, the majority of audio energy in human speech falls in the range of 200 Hz to 2500 Hz. Thus, transmissions that focus on the human voice, such as telephone, AM radio, and CBT programmes, can be limited to 4 kHz (Table 3.)

The use of audio in CBT can have negative effects. If the multisensory capabilities are not combined correctly, then the material may overload the student's capacity to receive and process information. Poorly designed courseware can be so complex as to confuse, frustrate or alienate the user. Courseware producers need to take into account how information is communicated effectively, and avoid undesirable student responses which have been caused by counterproductive techniques.

AUDIO SYSTEM	DYNAMIC RANGE (dB)	FREQUENCY RANGE
Telephone	35	100 Hz - 4 kHz
AM Radio	45	40 Hz - 4 kHz
FM Radio	60	20 Hz - 16 kHz
LP Records	70	20 Hz - 22 kHz
Compact Discs	95	20 Hz - 22 kHz
igital Audio (PC)	variable	20 Hz - variable
Human Ear	120	25 Hz - 16 kHz

Table 3. Dynamic and Frequency Ranges

Research into the best "mix" of audio, text, and motion has to date been led by the television industry. The advertising media spend large amounts on refining their products in order to achieve the highest memory retention levels which will result in consumers buying particular products. The same methods can be applied to CBT as the goals are identical - the viewer/student remembering the salient points and executing a proceedure with the information gained. The effects of audio and video applied in combination have been found to be different than when each medium is presented alone [2]. Experiments proved that the combined media results are not the simple sum of the individual media. Research has shown [3] that use of redundant information greatly enhances the learning process. For example, when an audio file is being played, bulleted text containing key words should appear on the screen in synchronisation. Audio files should not be made too long; there is a limit to the amount of redundancy which can be absorbed by the student and a lengthy audio section will detract from the learning experience.

As a general rule it has been found that students retain 20% of audio generated material, against an 80% retention level for totally visual material. Consequently, the run time of the audio segments should not be out of proportion to the graphic displays. This "rule" goes some way in explaining why the human brain is receptive to apparent redundancy. Experiments conducted by Drew and Grimes [4] found that greater audio and video redundancy led to a greater audio recall and increased understanding of content. This has been found true with recent courseware produced for Navy applications where the students have consistantly reported the desire to have supporting text displayed along with the audio segments.

ALTERNATIVE METHODS FOR INCORPORATING AUDIO

The conventional method of creating digital audio files is both time consuming and costly, especially in memory terms as discussed earlier. The use of text to speech converters appears to offer a way forward which could have significant advantages. Text to speech converters are not a new idea, but their development has been rather slow. The early converter systems produced crude robotic sounding audio, and while advances with text manipulating software have been made their use has been mainly in the field of automated telephone voice messaging. More recently, text to speech software has been incorporated with some conventional digital audio cards, but this has been aimed more at the domestic market for home multimedia use.

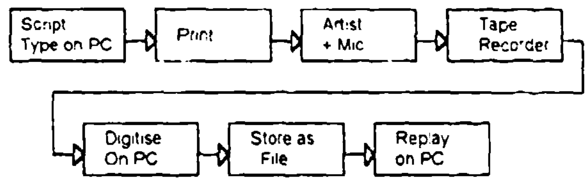


Figure 1 - Typical digital audio process

The process by which digital audio files are produced with conventional methods is shown in Figure 1. Very few organisations have their own in-house recording artists and this can lead to delays in the production programme when the received tapes are reviewed and modifications are required. There can be much reworking of the master tape, especially where the courseware has a high specific technical content alien to the recording artist's experience.

The text to speech production process is a great deal simpler and totally eliminates the requirement for a recording artist. The lesson script is typed up in the usual manner on a PC, with the file being saved in standard ASCII format. Typical converters can directly read the ASCII files and replay them as audio in real time. Thus one of the main advantages of this method is the ability to rapid prototype lesson material. Because the audio files are completely synthetically produced, courseware can be assembled directly from text files to give an immediate lesson draft. This is especially important where large volumes of training material are being produced and individual lesson authors require an early idea of the run times of the audio files. Text to speech systems also offer the same basic advantages percieved when CBT was first introduced- a consistant training medium, the Digital Speech Processor (DSP) does not have bad days, coughs or colds.

The memory savings obtained from using a text to speech system are quite outstanding. For example, a selected portion of audio used in a Navy CBT lesson consisted of 12 sentences or statements. The files used 1.167 Mb of audio memory after digitising with a conventional system, yet the ASCII file size only occupied 901

bytes. The perceived compression of 1295 : 1 is clearly phenominal and warrants further investigation.

Figure 2 - Text to Speech process

The disadvantage of current systems is that the default settings of the DSP produce a distinctly mechanically sounding voice. In the same way that the digital audio editor can cut and paste the human produced scripts, the DSP allows the user to fine tune the reconstituted files. For example, algorithms can set the syllable length, speech rate, pitch, and intonation. Pronunciation accuracy has reached over 99.4% with some voice messaging based systems. errors remaining can be easily corrected by the courseware author. Major features of a DSP system should include alphabetic pronunciation, punctuation timings, digit and abbreviation processing. A simple test can be applied to determine the capabilities of a text to speech system such as "Does Dr. Watson live at 221B Baker St?" A good system will have a default that recognises Dr for Doctor as opposed to Drive, and St for Street vice Saint. However, if the test is repeated with "Baker Dr.", the system should recognise "Baker Drive" and replay accordingly, with "221B" being replayed as usually said in conversation - "2-2-1-B". The better the DSP programming, the less work there will be for the author, and the faster the lessons will be produced. Thus the more care that is taken in setting up the initial default values, the greater the efficiency of courseware production.

FUTURE RESEARCH AREAS

There is an urgent need for detailed research into the benefits of incorporating audio into CBT. Although some studies have indicated a low retention rate for audio as opposed to visual information, other studies have not identified a significant difference [5,6.]. Determination of the ideal audio/visual combination has still to be studied, and the optimal A/V ratio may well be found to differ with different types of courseware [7], age and experience of the students.

Research is also needed in the area of optimal sampling rates and compression ratios for digital audio. Because of the storage space constraints, guidelines should be developed to help minimise storage without adversely impacting the students' achievement. In addition, comparison studies between the typical digital audio process and the text-to-speech process could help instructional designers select the most appropriate procedure for each application.

CONCLUSIONS

Hardware and software are now available to incorporate audio into CBT multimedia instructions. Along with the opportunity provided by this instructional medium, there are challenges to ensure that audio is implemented in the optimal manner. Through research and practice, developers of interactive courseware will be able to optimise the incorporation of digital audio in order to effectively increase the total learning experience.

REFERENCES

1. Barron, A.E. & Varnadoe, S.V. (1992). Digital Audio: A sound design element. Instruction Delivery Systems. 6(1), 6-9.

2. Travers, R.M.W. Research and Theory Related to Audiovisual Information Transmission (Rev. Ed.) US Department of Health, Education and Welfare, Office of Education, 1967.

3. Hartman, F.R. Recognition Learning Under Multiple Channel Presentation and Testing Conditions, A.V. Communication Review, 1961, a9, 24-43.

4. Drew, D.G. & Grimes, T. Audio-Visual Redundancy and TV News Recall, Communication Research, 1987, 14, 452-461.

5. Barron, A.E. & Kysilka, M. (1992). Digital audio for multimedia. Proceedings from the Ninth International Conference on Technology and Education, Paris, France, March 16 - 20.

6. Barron, A.E. & Atkins, D. (1992). The effects of textual redundancy in audio computer based education. Paper presented at the Florida Educational Research Association, 37th Annual Conference, Orlando, Florida, USA, November 11 - 14.

7. Hall, J.H. & Ralph, M.T. The Utilisation of ICW for Navy Training Systems. Proceedings from 3rd International Training Equipment Conference, Luxembourg, 7 - 9 April 1992.

STORYBOARD DEVELOPMENT FOR INTERACTIVE MULTIMEDIA TRAINING

Kay L. Orr, Ph.D.
Katharine C. Golas, Ph.D.
Southwest Research Institute
San Antonio, Texas

Katy Yao, Ph.D.
Consultant

ABSTRACT

Training applications using interactive multimedia capabilities are growing in number. The approach followed to produce these multimedia applications is essentially the same (analysis, design, development, implementation, and evaluation) regardless of the instructional delivery system.

Data from research studies, combined with development experience, provides insight into "what works best" for this particular delivery system, thus producing the most effective multimedia training in the most efficient manner. This paper addresses the procedures for storyboard development and provides specific guidelines for designing interactive multimedia courseware. Guidelines are presented for increasing interactivity, determining extent of learner control, determining most appropriate use of feedback, preparing visual elements (video, text, graphics and animation), audio elements, and programming. All of the guidelines are based on data from research studies. The research studies and literature which support the guidelines are specified by topic in the references.

ABOUT THE AUTHORS

Dr. Kay Orr is a research analyst in the Instructional Systems Section at Southwest Research Institute. She has worked on large-scale interactive multimedia projects, and she recently designed a workshop for effective preparation of storyboards for interactive courseware programs. She is an expert in the design and development of Digital Video Interactive programs. Her MS and PhD are in Instructional Technology from the University of Texas at Austin.

Dr. Katharine C. Golas is manager of the Instructional Systems Section at Southwest Research Institute. She began her career in ISD in 1977, by using the Interservice Procedures for Instructional Systems Development Model to develop print-based exportable job training packages. During the past 16 years, she has directed over 75 ISD projects, including twenty interactive videodisc projects and ten Digital Video Interactive (DVI)®* projects. She is currently directing research and development efforts using advanced multimedia training technologies. In 1992, she led a project team to redesign the Air Force ISD model and methodology. She has a PhD and MA in Instructional Systems from Florida State University.

Dr. Katy Yao is a consultant for Southwest Research Institute. She is an instructional development specialist and has served clients in the military, private industry, and education. She has managed and authored numerous training manuals and designed DVI®-based training systems. She recently designed a workshop for effective preparation of storyboards for interactive courseware programs. She holds MS and PhD degrees in Instructional Systems Technology from Indiana University.

---

*®DVI is a registered trademark of Intel Corporation.

STORYBOARD DEVELOPMENT FOR INTERACTIVE MULTIMEDIA TRAINING

Kay L. Orr, Ph.D.
Katharine C. Golas, Ph.D.
Southwest Research Institute
San Antonio, Texas

Katy Yao, Ph.D.
Consultant

BACKGROUND

A storyboard is the documentation for interactive multimedia production which contains instructions for programming, an audio script, and a detailed description of the visual elements such as text, video, graphics, and animation. It is typically developed by instructional designers, with input from other development team members such as subject matter experts, videographers, programmers, and graphic artists. Storyboards are developed during the design phase of the instructional systems development process. The storyboard becomes the key design document that the entire production team uses as a base for developing the interactive program. The storyboard information is often reviewed and approved by the customer prior to the start of the development effort.

This paper provides specific guidelines for storyboard development and the rationale, based on research findings, for each guideline. The research studies and literature which support each guideline are presented by topic in the references. It is unlikely that in any one program, every guideline will be implemented. The guidelines are not meant to be applicable to all situations and environments. Their application depends on factors such as the hardware and software selected for ICW development and delivery, the learning skills and motivation of the target audience, the complexity and criticality of the instructional content, and, of course, available resources. The guidelines should be adjusted based on these factors.

BASIC ICW INSTRUCTIONAL STRATEGIES

Instructional strategies are the general instructional treatment given to lessons in an interactive multimedia course. When developing storyboards, the designer will be concerned with ensuring that:

- Interactivity is increased.
- Learner control is addressed.
- Feedback is appropriate for enhancing learning and transfer.

GUIDELINES FOR INTERACTIVITY

In any type of computer-based training, interactivity refers to the activities performed by both the learner and the computer. The quantity of interaction depends on a number of variables, including the type of input required by the learner, how the response is analyzed, and how the computer responds back to the learner. Research has shown that it is important to design as much meaningful interactivity as possible into an ICW program (Hannafin, 1989, Lucas, 1992, Thompson and Jorgensen, 1989, Schwier and Misanchuk, 1988). Borsook (1991) argues that in order for interactive instruction to be truly interactive, it should emulate interpersonal communication. Guidelines for increasing interactivity in ICW programs are presented below.

1. Provide opportunities for interaction at least every three or four screens or, alternatively, about one per minute. However, mandatory interaction with the computer should not be superficial. Without interaction, the program is just a fancy electronic page turner. However, if an action required is somewhat superficial, the student may be distracted by it and become annoyed. Students prefer not to have superficial interaction.

2. Chunk the content into small segments and build in questions (with feedback), periodic

reviews, and summaries for each segment. Chunking content into smaller units and providing opportunities for interaction (e.g., questions) within each information segment allows students to interact with the program more frequently. "Blending" instruction with practice reduces boredom and at the same time facilitates learning.

3. Ask as many questions as possible without interrupting the continuity of the instructional flow. Questions provide information for the system to evaluate student performance and branch them to an appropriate place in the instruction. Questions also sustain student attention by keeping them involved in the learning process.

4. Ask a question after, but not immediately following, the related content. Sometimes a gap between a question and its related content will facilitate learning by forcing the learner to mentally search for and review necessary information, rather than requiring them to immediately repeat what they were just taught. This searching and reviewing process can enhance retention.

5. Ask students a question that they can figure out the answer to from previously learned knowledge. A straightforward presentation of new content can be boring.

6. Ask students to apply what they have learned rather than memorize and repeat answers.

7. Use rhetorical questions during instruction to get students to think about the content or to stimulate their curiosity. Also use them as a natural transition between frames. A rhetorical question does not require students to overtly provide an answer. It invites students to *mentally* interact with the content. Used as a transition aid, it can direct students' attention to what is coming up next.

8. Consider designs where the learner is not presented with information in a linear format, but rather discovers information through active exploration in the program. With some tasks, such as problem solving, learning through discovery promotes understanding and remembering because new knowledge is linked substantively and nonarbitrarily to existing knowledge.

GUIDELINES FOR LEARNER CONTROL

Learner control refers to the degree to which learners are allowed to take charge of the instruction and their learning environment: what to learn and how to learn it. In many instances, learners can make appropriate decisions about the most effective way to proceed through a training program. Research suggests, however, that in some instances, learners do not choose the most effective route (Chung and Reigeluth, 1992). Careful consideration of learner control issues is important in ICW design. Guidelines for learner control of sequence and content for ICW programs are presented below.

1. Provide learner control of **sequence** when:

   a. Lengthy instructional sequences must be completed by the student in no specific order. Student motivation and interest will be maintained because students will be in control and not forced through a particular sequence which ultimately does not affect learning.

   b. Students are familiar with a topic and are able to make appropriate sequence choices. In this case, motivation is facilitated because students can choose information that is interesting and relevant to them.

   c. The training is for cognitive strategies or higher-order problem solving tasks. Sequence control in this instance will allow students to make selections that may facilitate flexible and novel thinking.

2. Do **not** provide sequence control to students in situations where the materials have a specific prerequisite order. Learning could be inhibited if the sequence is improperly chosen.

3. Provide learner control of **content** when:

   a. Students have significant previous knowledge of the content. Presentation of known materials is irrelevant and often uninteresting to students.

   b. Students have higher ability (that is, they are "sophisticated" learners). Sophisticat-

ed learners are often able to make content choices based on their particular needs.

c. There is a high probability that students will succeed in learning the content regardless of the chosen content. Students will perceive through feedback that success is under their personal control and is relatively independent of the chosen content.

d. Cognitive strategies and higher-order problem-solving (rather than facts) are being taught. Students may see the relevance of different content and will be able to use this information effectively in novel ways during the learning of cognitive strategies and higher-order problem solving.

e. The skills are not critical, the training is optional, and student motivation is high.

4. Do **not** provide full learner control of content when all topics in the instructional presentation are required for successful completion of the program and there is a hierarchical order to the materials. If there is no hierarchical order to the lessons, let the students have control of the order but make sure they don't skip any relevant information.

5. Determine the amount of learner control based on your resource availability as well as these guidelines. Increased learner control over sequence and content generally requires more development work and more resources.

GUIDELINES FOR FEEDBACK

Feedback tells the learner about the accuracy of their response. Feedback can be used to address possible student misconceptions or lack of prerequisite knowledge. It can also be used to help students learn, enhance retention, and measure how much they have learned. Guidelines for feedback are presented below.

1. Keep feedback on the same screen with the question and student response. This reduces the memory load for the student.

2. Provide feedback immediately following a student response. Information about test results is important in the learning process. Delayed feedback can confuse students.

3. Provide feedback to verify the correctness and explain why. It may not be clear to students why their responses are correct or incorrect. Therefore, in addition to knowledge of results, feedback should provide specific information about a response.

4. For incorrect responses, give the student a hint and ask the student to try again. Without the hint, students may fail again and feel frustrated. The hint helps students recall relevant information to answer the question.

5. Tailor the feedback to each learner's response. Feedback should address the misconception a student may have by selecting a particular incorrect response.

6. Provide encouraging feedback. However, do not provide the type of feedback that may encourage a student to make an incorrect response on purpose just to see the feedback. Positive feedback can provide students with the motivation to learn. Cynical or negative feedback may discourage a student.

7. Add instructional feedback to simulation responses to explain why the simulated world reacted in a certain way or to provide a hint. In simulation, feedback is embedded in how the simulated world responds to a particular learner action. In the test, feedback can be phased out to facilitate transfer.

8. If possible, allow students to print out their test results. Students often like to maintain a hard copy record of their performance.

GUIDELINES FOR VISUAL ELEMENTS

Visual information in an ICW course serves to enhance the effectiveness of the training program. Visual elements include still frame and motion video, photographs, text, graphics, and animation. Guidelines for visual elements of an ICW program are presented below.

1. Do not jam a screen with too much information at any one point. Cluttered screens reduce learning efficiency and effectiveness (i.e., it takes more time to learn and more students often make more errors.)

2. When presenting a large amount of relevant information, display small chunks of information one at a time through:
   - Screen build-up
   - Window overlay
   - Icon buttons

3. Use windows to group or separate certain information from the rest of the display. This guideline helps to:
   - Draw students' attention to a particular set of data.
   - Reduce the density of display on the screen by superimposing one display on top of another.
   - Establish student expectancy that certain data will always appear in a certain format and location.

4. Use icon buttons for concrete concepts that can be represented pictorially in miniature. Icon buttons represent information that is available in a compact, easy-to-understand, pictorial format; and upon request of a student, disclose that information.

5. Consider presenting information graphically and spatially (e.g., in a diagram or a flowchart). Relationships among content or the overall program structure can be more easily visualized and remembered. A student's path through the program can be easily displayed and remembered.

6. Use the following techniques to keep students oriented:
   - Place certain information in constant locations.
   - Provide a consistent layout for the same types of screens.
   - Maintain the same perspective in a series of visuals. If a change of perspective is necessary, cue students to the change.
   - Use type sizes, colors, and shapes as cues.
   - Provide signposts which help a student know current and past locations, what lies ahead, and how to get there. Make signposts available for reference without requiring the student to move from the current location.
   - Provide a bird's-eye view, or long shot, before zooming into details, to establish a frame of reference for the student.

   Knowing where they are, how they got there, what they can do, where they can go and how they can get there gives students a sense of control. Making this information available allows students to concentrate on the program content rather than the navigation mechanism.

7. Use the following techniques to position information on a screen:
   - Present key information in prominent areas (e.g., away from the border).
   - Present information that changes from display to display (the body of the instruction) in the center of the screen.
   - Present recurrent information (e.g., menu bars) in constant locations.
   - Present navigation buttons near the borders of the screen.

8. To differentiate key information and attract or direct a student's attention, implement these cuing techniques:
   - Arrows, labels, narration
   - Separation of information into distinct objects
   - Windows
   - Colors, shapes
   - Highlighting, bordering, underlining
   - Mixed type sizes and fonts
   - Blinking

9. Use the following techniques for cuing information:
   - Reserve blinking for critical situations requiring immediate student attention or action.
   - Keep borders distinct from the object enclosed.
   - Highlight by either brightening the area of interest or dimming the background.
   - Limit highlighting to 10 percent of the display for effectiveness.
   - Avoid using too many cues at one time. Oversaturation of the techniques may reduce their effectiveness.

10. Use the following techniques for colors:
    - Limit the number of colors on each display. Too many colors on a display reduce effectiveness and aesthetic quality.
    - Use black on yellow, or black on white for text. Always use dark letters on a light background. Blue is an excellent back-

ground color. But don't use blue for text, edges, narrow lines, or small objects.

- Avoid distinctions based on the color cue only. When using colors, always use a second cue (e.g., label, shape, texture) for color-blind students.

GUIDELINES FOR MOTION VIDEO

Motion video is often a major element of ICW. A high level of detail is necessary in the storyboard to ensure that the video producer has sufficient information to get an accurate video shot. Guidelines for motion video are presented below.

1. Present all information in three-shot sequences (long, medium, and close-up) to establish visual orientation. Use close-up shots to grab the student's attention and imply that something is important. Use long shots to establish frames of reference. Try to avoid static shots when shooting motion video.

2. Use a zoom-in to focus a student's attention on a particular object while maintaining visual orientation. This provides a similar effect to a three-shot sequence.

3. When showing something new, focus on the subject long enough for the audience to register what is being shown. Once the audience has seen the subject in the shot, you don't have to focus on it as long the next time you show it.

4. Keep the main subject well lit and watch for possible background distractions. The eye focuses on lighted instead of dark areas and movement instead of static images.

5. Consider using the following motion video formats:
   - Facility/event walk-through (with an off-screen narrator)
   - Lecture (talking head)
   - Demonstration (show and tell) and modeling
   - Interview
   - Talk show format
   - Panel discussion
   - Dramatization
   - Simulation

6. Use "first-person" simulation to allow the student to perform actions as closely as possible to the actual situation (e.g., operating a piece of equipment or troubleshooting). Usually first-person simulation is the preferred method because it facilitates transfer from training to on-the-job performance.

7. Use "third-person" or directed simulation to allow the student to vicariously experience the situation by directing a "person" in the program to perform whatever actions the student wants to perform. A "third-person" simulation may be more appropriate when you want the students to explore the consequences of both right and wrong behaviors in a high-risk situation.

8. Use audio and video to reinforce each other. Never present two unrelated or clashing pieces of information at the same time with audio and video. Design a visual message appropriate to the content and make sure that each visual ties in directly to the accompanying audio. Presenting unrelated or clashing information or an inappropriate visual will often confuse the student.

9. Present a series of visuals before or at the end of instruction. Quick visual inserts presented *before* instruction stimulate recall of prerequisites, serve as an advance organizer, direct attention to key information, and heighten interest. Quick visual inserts presented *after* instruction remind the audience of the key information and enhance retention.

10. Show future events or consequences of unacceptable performance (e.g., disaster caused by human errors) prior to instruction. This guideline is useful to impress the audience with the serious outcomes associated with unacceptable performance and to motivate the audience to adopt acceptable behaviors or practices.

11. Repeat program content in either an identical format or a different perspective to draw attention to particular items, heighten interest, and enhance retention. Things that are repeated are often remembered better. The mere fact that something is repeated implies that it is important.

12. Use motion video rather than still frame if the content requires movement to clearly depict

the point. Use still frames if production resources are limited or there are storage limitations with hardware.

Although expensive to produce, full-motion video can be used to represent reality and help the student achieve a high degree of transfer from training to on-the-job performance. Motion video can often add motivational value to training. For these reasons, motion video is often used to support affective domain objectives and simulations. However, it may be impractical or impossible to produce full-motion video. If this is the case, animation sequences and graphics may be substituted so that instructional effectiveness is not compromised.

GUIDELINES FOR GRAPHICS/ANIMATION

Graphics and animation sequences are often developed to enhance learning. Guidelines for graphics and animation design are presented below.

1. Use graphics or animation when:
   - A realistic presentation (i.e., video) may overwhelm the audience with too much detail.
   - Conditions or problems to be portrayed occur so infrequently that a video presentation is not practical.
   - Minute details are required. Video often has lower resolution than graphics.

2. Use graphics to reduce irrelevant details and highlight key information. Video may be used together with or following the graphic presentation.

3. Avoid biases or stereotypes in graphics or animation (gender, ethnic groups, etc.). Use of biases or stereotypes is insulting and distracting.

4. Use exaggeration and humor carefully to heighten student interest and to facilitate recall. People often remember exaggerated or humorous information better and can be motivated by it.

GUIDELINES FOR TEXT

Text is often used to present content or highlight certain information. Guidelines for designing text are presented below.

1. Limit the amount of text on screen. It is more difficult and takes longer to read text on a screen than in print. People read text on a computer screen at a rate 28 percent slower than reading from a book.

2. Position text appropriately. Regular text should be left-justified only. Center headings and titles. Don't hyphenate words at the end of a line.

3. Use the following format techniques:
   - Provide generous white space to separate blocks of information.
   - Use headings as content summarizers and navigation aids.
   - Convert sentences containing serial items to lists.
   - Organize complex information into tables to help learners integrate program content.
   - Reserve use of all upper case for emphasis and titles only.

4. Use the following attention-getting techniques:
   - Limit highlighting or boldface to 10 percent of the display.
   - Use italic type for titles or headings.
   - Use reverse video or blinking with extreme discretion. Never blink text to be read.
   - Use mixed type sizes or fonts to differentiate screen components.
   - Use no more than one attention-getting technique on a single screen. Remember that oversaturation will reduce the effectiveness of these techniques.

5. Verify the appropriateness of the colors used for text under simulated presentation conditions. The clarity of colors used for text will vary depending on such factors as lighting of the room where the ICW stations are and proximity of the student to the machine.

GUIDELINES FOR AUDIO

The audio part of a storyboard is used by the narrator during audio production. Guidelines for audio design are presented below.

1. Use audio for primary presentation of the program content when the message is short, simple, and requires immediate student response; or if the target audience has poor reading skills.

2. Don't allow audio to interfere with reading from the text and vice versa. To be most effective, audio and text should complement, not compete with, each other.

3. Don't put a lot of text on a single screen. Research data indicates that students find it easier to complete lessons which use audio extensively to present information. Students generally prefer not to have to read long text passages off a screen.

4. Don't let audio compete with video presentations. Audio should support rather than contradict or interfere with visuals. Long silences or competing audio and video may confuse students.

5. If audio is used, provide students with headphones. Students in a lab environment will not be distracted by the audio from other student stations if headphones are provided.

6. When scripting narration, consider using the following techniques:
   - Visualize the images that will be presented on the screen during the narration.
   - Use style and tone appropriate to students' language ability, subject matter knowledge, and vocabulary.
   - Write the script for the ear, not the eye. Read the script out loud to yourself and listen to how it sounds.
   - Keep the language simple, use the active voice, and be direct.
   - Use short sentences.
   - Watch out for acronyms, technical jargon, and unfamiliar terms. Define them if you have to use them.
   - Make the transitions from one concept to another clear.
   - Provide a corresponding visual for every piece of narration.
   - Avoid long pauses in visuals while waiting for extended narration to finish.
   - Select appropriate narrators.
   - Alternate male and female voices to provide variety and maintain audience attention.

7. To make it easier for the narrator or professional talent to record or read the ICW audio, use the following techniques:
   - Number all pages in the upper right-hand corner.
   - Use a legible type size.
   - Specify how acronyms should be read.
   - Spell out all numbers.
   - Spell difficult words and names phonetically.
   - Separate each letter in an abbreviation with a hyphen (e.g., I-C-W).
   - Describe nonverbal cues in parentheses.
   - Indicate pauses by the word "PAUSE" in parentheses.
   - Indicate emphases in parentheses if inflection is not obvious.
   - Double or triple space between lines.

8. Stick to the message. Tell the students only what is relevant.

9. Keep the audio script short and simple. If the message is too long, break it into chunks separated by instructional activities (e.g., quizzes, reviews, hands-on exercises). Students may get bored if they receive information passively from the program for an extended period of time.

10. Use sound effects as cues. Once the link between a sound effect and a specific event is established, the sound effect can serve as an efficient navigation aid, such as the following:
    - Use a beep or an "oh-oh" to clue students that they've done something incorrectly on the screen (e.g., wrong entry). Provide headphones to the students so classmates won't know when mistakes are being made.
    - Use tunes associated with certain events in the program (e.g., introduce a quiz with a short music sequence).

11. Keep production limits in mind (i.e., budget, time, and technical capabilities of production staff and equipment). Allow time for audio rework, which could happen as the development effort proceeds. You obviously want to avoid reaching a point in the development effort where you have run out of funds and "aren't quite finished" with the program.

GUIDELINES FOR PROGRAMMING

The actual programming or authoring of an ICW program typically occurs during the development phase. However, consideration needs to be given to a number of programming issues during

storyboard design. It is wise to establish programming standards before beginning to storyboard the content. Standards save time; they eliminate the need for reinvention and modification. Standards also promote clarity and consistency. Although a certain degree of flexibility is necessary and changes may occur along the way, standards establish consistency throughout the entire ICW program. Follow these program standards, unless you can offer a convincing argument as to why the standards are not applicable to your design.

Consider Programming Standards or Conventions for:

Screen Type

- Course/lesson/subject title screen
- Introduction/overview screen
- Instructional screen
- Inserted question and feedback screen
- Review screen
- Summary screen
- Practice/exercise screen
- Test screen
- Help screen

Screen Layout

- Amount of text
- Text placement
- Headings
- Margins
- Text font and size
- Captions
- Color (text, background, emphasis, borders)
- Attention-getting cues
- Paragraph indentation
- Buttons (what - navigation/help/content; format - icon/text)
- Menus (structure, labels)
- Windows

Questions and Feedback

- Presentation of questions (text, audio, graphics, or combination)
- Type of student responses required (pointing, selecting, or text entry)
- Number of tries allowed
- Hints
- Type of feedback for each try (knowledge of result, explanation, remediation)
- Presentation of feedback (text, audio, graphics, or combination)

Presentation Sequence in Each Segment

- Title screen
- Opening (motivational video segment)
- List of objectives
- Main body of instruction with inserted questions and periodic reviews
- Summary
- Exercise, practice, and test

Miscellaneous

- Naming conventions for video segments and files
- Transition
- Sign-on procedures
- Cursor placement on each new screen
- Voice (e.g., referring to students as "you" and the program as "I" or a third person)
- Movement instruction (given via audio channel or buttons on the screen)

SUMMARY

The guidelines presented in this paper are based on over ten years of research regarding the design and development of interactive multimedia courseware. Because the research findings are at times contradictory, it is important that the guidelines and approaches that you select be based on the particular circumstances of your training application and users. For example, factors such as the learning skills and motivation of the target audience, the complexity of the instructional content, and the hardware and software selected for ICW development and delivery will greatly affect the courseware design. These guidelines are not meant to be applicable to all learning situations and training environments. The guidelines should be selected and adjusted based on specific program requirements and resources.

TOPICAL BIBLIOGRAPHY

Interactivity

Arwady, J., and Gayeski, D.M. (1989). *Using Video: Interactive and Linear Designs*. Englewood Cliffs, NJ: Educational Technology Publications.

Borsook, T. (1991). Harnessing the Power of Interactivity for Instruction. In M.R. Simonson and C. Hargrave (Eds.), *Proceedings of the 1991 Convention of the Association for Educational Communications and Technology* (103-117). Orlando, FL: Association for Educational Communications and Technology.

Brozo, W.G., Schmelzer, R.V., and Spires, H.A. (1983). The Beneficial Effect of Chunking on Good Readers' Comprehension of Expository Prose. *Journal of Reading*, 26, 442-445.

Carver, R.P. (1970). Effect of "Chunked" Typography on Reading Rate and Comprehension. *Journal of Applied Psychology*, 54, 288-296.

Hannafin, M.J. (1984). Guidelines for Determining Locus of Instructional Control in the Design of Computer-Assisted Instruction. *Journal of Instructional Development*, 7(3), 6-10.

Hannafin, M.J. (1989). Interaction Strategies and Emerging Instructional Technologies: Psychological Perspectives. *Canadian Journal of Educational Communications*, 18(3), 167-179.

Hannafin, M.J., and Colamaio, M.E. (1987). The Effects of Variations in Lesson Control and Practice on Learning from Interactive Video. *Educational Communications and Technology Journal*, 35(4), 203-212.

Jonassen, D.H. (1985a). Interactive Lesson Designs: A Taxonomy. *Educational Technology*, 25(6), 7-17.

Jonassen, D.H. (1985b). Interactive Lesson Designs: A Taxonomy. *Educational Technology*, 25(7), 7-17.

Litchfield, B.C. (1990). Criteria for Evaluating Videodisc Interactivity. *Performance and Instruction*, 29(6), 23-26.

Lucas, L. (1992). Interactivity: What Is It and How Do You Use It? *Journal of Educational Multimedia and Hypermedia*, 1(1), 7-10.

Mattoon, J.S., Klein, J.D., and Thurman, R.A. (1991). Learner Control Versus Computer Control in Instructional Simulation. In M.R. Simonson and C. Hargrave (Eds.), *Proceedings of the 1991 Convention of the Association for Educational Communications and Technology* (481-498). Orlando, FL: Association for Educational Communications and Technology.

Novick, M.R., and Lewis, C. (1974). *Prescribing Test Length for Criterion-Referenced Measurement* (Tech Bull. No 18). Iowa City, IA: American College Testing Program.

Salisbury, D.F., Richards, B.F., and Klein, J.D. (1985). Designing Practice: A Review of Prescriptions and Recommendations from Instructional Design Theories. *Journal of Instructional Development*, 8(4), 9-19.

Schaffer, L.C., and Hannafin, M.J. (1986). The Effects of Progressive Interactivity on Learning from Interactive Video. *Educational Communications and Technology Journal*, 34, 89-96.

Schneiderman, B. (1987). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Reading, MA: Addison-Wesley.

Schwier, R.A., and Misanchuk, E.R. (1988). The Effect of Interaction and Perceived Need for Training on Learning from Computer-Based Instruction. *Canadian Journal of Educational Communication*, 17(3), 147-158.

Smith, S.L. and Mosler, J.N. (1984). *Design Guidelines for the User Interface for Computer-Based Information Systems*. The MITRE Corporation, Bedford, MA, 01730, Electronic Systems Division. Cited in Schneiderman, B. (1987). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Reading, MA: Addison-Wesley. (Available from the National Technical Information Service, Springfield, VA.)

Thompson, J.G., and Jorgensen, S. (1989). How Interactive is Instructional Technology? Alternative Models for Looking at Interactions Between Learners and Media. *Educational Technology*, 29(2), 24-26.

Learner Control

Arnone, M.P., and Grabowski, B.L. (1991). Effect of Variations in Learner Control on Children's Curiosity and Learning from Interactive Video. In M.R. Simonson and C. Hargrave (Eds.), *Proceedings of the 1991 Convention of the Association for Educational Communications and Technology* (45-67). Orlando, FL: Association for Educational Communications and Technology.

Carrier, C.A., and Jonassen, D.H. (1988). Adapting Courseware to Accommodate Individual Differences. In D.H. Jonassen (Ed.), *Instructional Designs for Microcomputer Courseware*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Chung, J. and Reigeluth, C.M. (1992). Instructional Prescriptions for Learner Control. *Educational Technology*, October, 14-19.

Duchastel, P.C. (1986). Intelligent Computer-Assisted Instruction Systems: The Nature of Learner Control. *Journal of Educational Computing Research*, 2(3), 379-393.

Gay, G. (1986). Interaction of Learner Control and Prior Understanding in Computer-Assisted Video Instruction. *Journal of Educational Psychology*, 78, 225-227.

Higginbotham-Wheat, N. (1988). *Perspectives on Implementation of Learner Control in CBI*. Paper presented at the Annual Meeting of the Mid-South Educational Research Association, Lexington, KY, November (ERIC Document Reproduction Service No. ED 305 898).

Higginbotham-Wheat, N. (1990). Learner Control: When Does it Work? In M.R. Simonson and C. Hargrave (Eds.), *Proceedings of the 1990 Convention of the Association for Educational Communications and Technology*. Anaheim, CA: Association for Educational Communications and Technology (ERIC Document Reproduction Service No. ED 323 930).

Kinzie, M.B., Sullivan, H.J., and Berdel, R.L. (1988). Learner Control and Achievement in Science Computer-Assisted Instruction. *Journal of Educational Psychology*, 80(3), 299-303.

Laurillard, M. (1984). Interactive Video and the Control of Learning. *Educational Technology*, 24(6), 7-15.

Merrill, M. D. (1984). What is Learner Control? In R.K. Bass and C.D. Dills (Eds.), *Instructional Development: The State of the Art II*. Dubuque, IA: Kendall/Hunt.

Milheim, W.D., and Azbell, J.W. (1988). How Past Research on Learner Control Can Aid in the Design of Interactive Video Materials. In M.R. Simonson and J.K. Frederick (Eds.), *Proceedings of the 1988 Convention of the Association for Educational Communications and Technology* (459-472). New Orleans, LA: Association for Educational Communications and Technology (ERIC Document Reproduction Service No. ED 295 652).

Ross, S., Morrison, G., and O'Dell, J. (1990). *Uses and Effects of Learner Control of Context and Instructional Support in Computer-Based Instruction*. Paper presented at the Annual Meeting of the Association of Educational Communications and Technology, Anaheim, CA, February.

Santiago, R.S., and Okey, J.R. (1990). *Sorting Out Learner Control Research: Implications for Instructional Design and Development*. Paper presented at the Annual Conference of the Association for Educational Communications and Technology, Anaheim, CA, February.

Steinberg, E.R. (1977). Review of Student Control in Computer-Assisted Instruction. *Journal of Computer-Based Instruction*, 3(3), 84-90.

Tennyson, R., Park, O., and Christensen, D. (1985). Adaptive Control of Learning Time and Content Sequence in Concept Learning Using Computer-Based Instruction. *Journal of Educational Psychology*, 77(4), 481-491.

Feedback

Dalton, D.W., and Hannafin, M.J. (1987). Examining the Effects of Varied Computer-Based Reinforcement on Self-Esteem and Achievement: An Exploratory Study. *Association for Educational Data Systems Journal*, 18(3), 172-182.

Frick, T.W. (1989). Bayesian Adaptation During Computer-Based Tests and Computer-Guided Practice Exercises. *Journal of Educational Computing Research*, 5(1), 89-114.

Frick, T.W. (1990). A Comparison of Three Decision Models for Adapting the Length of Computer-Based Mastery Tests. *Journal of Educational Computing Research*, 6(4), 479-513.

Schloss, P.J., Wisniewski, L.A., and Cartwright, G.O. (1988). The Differential Effect of Learner Control and Feedback in College Students' Performance on CAI Modules. *Journal of Educational Computing Research*, 4(2), 141-149.

Visual Elements

Anglin, G.J. and Kwak, E. (1991). *Research on Pictures: Knowledge Acquisition, Visual Thinking, and Cognitive Skill Acquisition: A Guide to the Literature, 1986-1990*. Paper presented at the Annual Convention of the Association for Educational Communications and Technology, Orlando, FL, February.

Apple Computer, Inc. (1987). *Human Interface Guidelines: The Apple Desktop Interface*. Reading, MA: Addison-Wesley.

Aspillaga, M. (1991). Implications of Screen Design Upon Learning. *Journal of Educational Technology Systems*, 20(1), 53-58.

Berry, L. (1991). Visual Complexity and Pictorial Memory: A Fifteen Year Research Perspective. In M. R. Simonson and M. Treimer (Eds.), *Proceedings of Selected Research Paper Presentations at the 1985 Annual Convention of the Association for Educational Communications and Technology* (92-102). Ames, IA: Iowa State University Press.

Braden, R. (1986). Visuals for Interactive Video: Images for a New Technology. *Educational Technology*, 26(5), 18-23.

Dwyer, F. M. (1978). *Strategies for Improving Visual Learning*. State College, PA: Learning Services.

Fleming, M., and Levie, W.H. (1978). *Instructional Message Design*. Englewood Cliffs, NJ: Educational Technology Publications.

Grabinger, R.S. (1989). Screen Layout Design: Research into the Overall Appearance of the Screen. *Computers in Human Behavior*, 5, 175-183.

Hannafin, M.J., and Hooper, S. (1989). An Integrated Framework for CBI Screen Design and Layout. *Computers in Human Behavior*, 5, 155-165.

Hathaway, M.D. (1984). Variables of Computer Screen Design and How They Affect Learning. *Educational Technology*, 24(1), 7-11.

Heines, J.M. (1984). *Screen Design Strategies for Computer-Assisted Instruction*. Bedford, MA: Digital Press.

Iuppa, N.V. (1984). *A Practical Guide to Interactive Video Design*. White Plains, NY: Knowledge Industry Publications.

Lucas, L. (1991). Visually Designing the Computer-Learner Interface. *Educational Technology*, July, 56-58.

Schneiderman, B. (1987). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Reading, MA: Addison-Wesley.

Smith, S.L. and Mosler, J.N. (1984). *Design Guidelines for the User Interface for Computer-Based Information Systems*. The MITRE Corporation, Bedford, MA 01730, Electronic Systems Division. Cited in Schneiderman, B. (1987). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Reading, MA: Addison-Wesley. (Available from the National Technical Information Service, Springfield, VA.)

Motion Video

Allen, B.S. (1986). A Theoretical Framework for Interactivating Linear Video. *Journal of Computer-Based Instruction*, 13, 107-112.

Anglin, G.J. and Kwak, E. (1991). *Research on Pictures: Knowledge Acquisition, Visual Thinking, and Cognitive Skill Acquisition: A Guide to the Literature 1986-1990*. Paper presented at the Annual Convention of the Association for Educational Communications and Technology, Orlando, FL, February.

Arwady, J., and Gayeski, D.M. (1989). *Using Video: Interactive and Linear Designs*. Englewood Cliffs, NJ: Educational Technology Publications.

Braden, R. (1986). Visuals for Interactive Video: Images for a New Technology. *Educational Technology*, 26(5), 18-23.

Copeland, P. (1988). Interactive Video: What the Research Says. *Media in Education and Development*, 2(2), 60-63.

DeBloois, M.L. (1982). *Videodisc/Microcomputer Courseware Design*. Englewood Cliffs, NJ: Educational Technology Publications.

Gayeski, D. (1983). *Corporate and Instructional Video Design and Production*. Englewood Cliffs, NJ: Prentice-Hall.

Pioneer Video. (1984). *Post-Production and Formatting Information*. Montvale, NJ: Pioneer Video.

Graphics/Animation

Anglin, G.J. and Kwak, E. (1991). *Research on Pictures: Knowledge Acquisition, Visual Thinking, and Cognitive Skill Acquisition: A Guide to the Literature 1986-1990*. Paper presented at the Annual Convention of the Association for Educational Communications and Technology, Orlando, FL, February.

Merrill, P.F., and Bunderson, C.V. (1981). Preliminary Guidelines for Employing Graphics in Instruction. *Journal of Instructional Development*, 4(4), 2-9.

White, J.V. (1983). *Mastering Graphics*. New York: Bowker.

Text

Duchnicky, R.L., and Kolers, P.A. (1983). Readability of Text Scrolled on Visual Display Terminals as a Function of Window Size. *Human Factors*, 25(6), 683-692.

Frase, L.T., and Schwartz, B.J. (1979). Typographical Cues That Facilitate Comprehension. *Journal of Educational Psychology*, 71, 197-206.

Garner, K.H. (1991). 20 Rules for Arranging Text on a Screen. In R.B. Frantzreb (Ed.), *Training and Development Yearbook, 1991 Edition* (4.16-4.18). Englewood Cliffs, NJ: Prentice-Hall.

Gillingham, M.G. (1988). Text in Computer-Based Instruction: What the Research Says. *Journal of Computer-Based Instruction*, 15(1), 1-6.

Glynn, S.M., Britton, B.K., and Tillman, M.H. (1985). Typographical Cues in Text: Management of the Reader's Attention. In D.H. Jonassen (Ed.), *The Technology of Text (Volume Two): Principles for Structuring, Designing and Displaying Text* (192-209). Englewood Cliffs, NJ: Educational Technology Publications.

Grabinger, R.S. (1984). CRT Text Design: Psychological Attributes Underlying the Evaluation of Models of CRT Text Displays. *Journal of Visual Verbal Languaging*, 4(1), 17-39.

Grabinger, R.S. (1985). *Relationships Among Text Format Variables in Computer-Generated Text*. Paper presented at the Annual Conference of the Association for Educational Communications and Technology, Research and Theory Division, Las Vegas, NV, January.

Grabinger, R.S., and Amedeo, D. (1985). CRT Text Layout: Prominent Layout Variables. In M.

R. Simonson and M. Treimer (Eds.), *Proceedings of Selected Research Paper Presentations at the 1985 Annual Convention of the Association for Educational Communications and Technology*, Ames, IA: Iowa State University Press.

Hartley, J. (1987). Designing Electronic Text: The Role of Print-Based Research. *Educational Communications and Technology Journal*, 35(1), 3-17.

Hooper, S. and Hannafin, M.J. (1986). Variables Affecting the Legibility of Computer-Generated Text. *Journal of Instructional Development*, 9(4), 22-28.

Isaacs, G. (1987). Text Screen Design for Computer-Assisted Learning. *British Journal of Educational Technology*, 1(18), 41-51.

Jonassen, D.H. (Ed.) (1982). *The Technology of Text: Principles for Structuring, Designing, and Displaying Text*. Englewood Cliffs, NJ: Educational Technology Publications.

Kang, T.J. and Muter, P. (1989). Reading Dynamically Displayed Text. *Behavior and Information Technology*, 8(1), 33-42.

Kolers, P.A., Duchnicky, R.L., and Ferguson, D.C. (1981). Eye Movement Measurement of Readability of C.R.T. Displays. *Human Factors*, 23(5), 517-527.

Kruk, R.S., and Muter, P. (1984). Reading of Continuous Text on Video Screen. *Human Factors*, 26(3), 339-345.

Mason, J.M., and Kendall, J.R. (1979). Facilitating Reading Comprehension Through Text Structure Manipulation. *The Alberta Journal of Educational Research*, 25, 68-76. Cited by O'Shea, L.J., and Sindelar, P.T. (1983), in The Effects of Segmenting Written Discourse on the Reading Comprehension of Low- and High-Performance Readers. *Reading Research Quarterly*, 18, 458-465.

Misanchuk, E.R. (1989). Learner Preferences for Screen Text Attributes in a Macintosh Microcomputer Learning Environment. In *Transitions: Proceedings of the AMTEC '89 Conference*. Edmonton, AB: Association for Media and Technology in Education in Canada.

Morrison, G.R., Ross, S.M., and O'Dell, J.K. (1988). Text Density Level as a Design Variable in Instructional Displays. *Educational Communications and Technology Journal*, 36(1), 103-115.

Muter, P., Kruk, R.S., Buttigieg, M.A., and Kang, T.J. (1988). Reader-Controlled Computerized Presentation of Text. *Human Factors*, 30(4), 473-486.

Ross, S.M., Morrison, G.R., and O'Dell, J.K. (1988). Obtaining More Out of Less Text in CBI: Effects of Varied Text Density Levels as a Function of Learner Characteristics and Control Strategy. *Educational Communications and Technology Journal*, 36(3), 131-142.

Audio

Buxton, W. (1989). Introduction to This Special Issue on Nonspeech Audio. *Human-Computer Interaction*, 4, 1-9.

DeBloois, M.L. (1982). *Videodisc/Microcomputer Courseware Design*. Englewood Cliffs, NJ: Educational Technology Publications.

Gaver, W.W. (1989). The SonicFinder: An Interface That Uses Auditory Icons. *Human-Computer Interaction*, 4, 67-94.

Grimes, T. (1990). Audio-Video Correspondence and Its Role in Attention and Memory. *Educational Technology Research and Development*, 38(3), 15-26.

Huber, D.M., and Runstein, R.E. (1989). *Modern Recording Techniques* (3rd Ed.). Carmel, IN: Howard W. Sams and Company.

Lehrman, P. D., and Tully, T. (1991). Catch a Wave: Digital Audio. *MacUser*, October, 94-103.

Watkinson, J. (1988). *The Art of Digital Audio*. London: Focal Press.

Programming

Barker, P.G. (1984a). MUMEDALA—An Approach to Multimedia Authoring. *British Journal of Educational Technology*, 15(1), 4-13.

Barker, P.G. (1984b). MUMEDALA: An Approach to Multi-Media Authoring. *Computers and Education*, 8(4), 463-469.

Bork, A. (1990). Practical Techniques Useful in Authoring Technology-Based Learning Material. *Journal of Computer-Based Instruction*, 17(2), 53-60.

Cook, E.K. (1990). The Use of Macintosh Authoring Languages in Effective Computer-Assisted Instruction. *Journal of Educational Technology Systems*, 18(2), 109-122.

Dean, C.T. (1988). Storyboarding for Computer-Based Training: A Technique in Transition. *Performance and Instruction*, 27(5), 8-14.

Hunka, S. (1989). Design Guidelines for CAI Authoring Systems. *Educational Technology*, 29(11), 12-17.

Merrill, M.D. (1985). Where is the Authoring in Authoring Systems? *Journal of Computer-Based Instruction*, 12(4), 90-96.

ENHANCEMENTS TO THE DISTRIBUTED INTERACTIVE SIMULATION ARCHITECTURE FOR TRAINING SIMULATOR INTEROPERABILITY

Brett E. Butler
Loral Advanced Distributed Simulation
Orlando, FL
(407)384-3014, butler@wdl1.wdl.loral.com

ABSTRACT

The Strawman version of the Distributed Interactive Simulation (DIS) Architecture was unveiled in March 1992. This Architecture addressed the requirements and design of interactive (man-in-the-loop) combat simulation in a distributed and networked computing environment.

Since its initial unveiling, work has continued on the refinement and expansion of the architecture. This paper highlights developments which, as part of the Architecture, facilitate the interoperation of training simulators of varied fidelity, design, and manufacture.

A specific application which motivates this work is the requirement to conduct training simulation exercises which utilize three different classes of networked simulator devices—a class of existing DIS trainers, an existing high-definition engineering simulator, and a class of new-generation DIS simulators.

Specific issues to be addressed in the paper are:

- *Summary overview of key concepts of the DIS Architecture and its enhancements.*
- *Brief comparison of the configurations and capabilities of both the existing "SIMNET" devices and the newer-generation training devices.*
- *Analysis of how simulator differences detract from the interoperation of these systems.*
- *Discussion of the concepts and solutions, found in the DIS Architecture, which address interoperability problems.*

The focus of the paper is to address how the Architecture supports the implementation of interoperability solutions in the proposed exercises.

ABOUT THE AUTHOR

Brett Butler, a Systems Engineer with Loral Advanced Distributed Simulation, has over ten years of experience in the fields of flight simulation, networked simulation, and software engineering. Mr. Butler has an M.S. degree in Mathematics from Arizona State University.

INTRODUCTION

The Strawman version of the Distributed Interactive Simulation Architecture (known as the "DIS Architecture") was unveiled at the Sixth Workshop on Standards for the Interoperability of Defense Simulations in March 1992. This Architecture addressed the design requirements of interactive, man-in-the-loop combat simulation in a distributed, networked computing environment. The purpose of the Architecture is as stated:

> "The DIS Architecture defines a time and space coherent representation of a virtual battlefield environment, measured in terms of the human perception and behaviors of warfighters interacting in free play. It provides a structure by which independently developed systems *(e.g. training simulators)* may interact with each other in a well managed and validated combat simulation environment..."

Since its initial unveiling, the DIS Architecture has been expanded and revised. This work has been supported by the Army's Battlefield Distributed Simulation—Developmental (BDS-D) program. This paper highlights the key characteristics of the revised Architecture with respect to interoperability. Interoperability is the goal of facilitating the joint use of training simulators of varied fidelity, resolution, design and manufacture.

The specific application which motivates the work described in this paper is the requirement to conduct training simulation exercises which utilize three classes of simulator environments: an existing class of networked simulator devices (the Simulator Networking, or SIMNET system), a high-definition engineering simulator at NASA Ames Crewstation Research and Development Facility (CSRDF), and a class of newer-generation simulators being developed as part of BDS-D. This combined system will support tactical combat training—teaching the specifics of coordination, cooperation, and teamwork on the combined-arms battlefield.

Specific issues to be addressed in the paper are:

- *Comparison of the configurations and capabilities of the different simulator training devices.*
- *Analysis of how these differences cause problems in the proposed interoperation of these systems.*
- *Discussion of the solutions that have been developed to meet these interoperability problems.*
- *Demonstration of how these solutions are incorporated into the DIS Architecture.*

ARCHITECTURE OVERVIEW

The DIS Architecture was first presented in Strawman form in March of 1992. Since that time it has been expanded and refined. Space will not allow a full explanation of the Architecture. However, we will present some key features that relate to interoperability.

Architecture Composition

In general, an architecture consists of a reference model (to establish common conception and discourse), and an attendant set of standards (to establish commonalty in design and interoperability). In the past, the focus of the DIS community has been on the message protocol between simulators. This message standard (formally known as "Standard for Information Technology—Protocols for Distributed Interactive Simulation Applications" [7]) is commonly known as the DIS Protocol. DIS Architecture developers decided that the DIS protocol, while necessary, did not go far enough. Additional standards were needed to ensure compatible visual, terrain, weapons, and dynamics models. The Architecture uses

the term "DIS Standards" for these additional standards specified by the Architecture.

Public vs. Private

Every architecture attempts to identify the boundary between private design and public conformance to a standard. The DIS Architecture is no exception. However, in DIS, with its emphasis on promoting interoperability between legacy simulation systems, this boundary line is even more important.

A tradeoff faced in this consideration is to weigh the promotion of comprehensive interoperability (by levying rigorous standards) against the accommodation of legacy systems. An architecture with light standards that accommodates legacy systems is said to be "non-invasive". As we shall see, the demands of useful interoperability under this application forced the architecture to be somewhat invasive.

Time and Space Coherence

Time and space coherence is the key objective of the DIS Architecture. Time and space coherence is not simulation fidelity. Fidelity describes how well the synthetic environment maps to reality. Time and space coherence is instead concerned with preservation of the simulation illusion and the maintenance of consistent experience and sensation for all simulation participants.

Implementation Principles

DIS technology is based on a core set of implementation principles which must be woven into the fabric of the architecture.

- *Autonomous simulation entities interacting in real time via networks using local copies of a common terrain and models database.*
- *Each DIS entity maintains its own world view, as a function of: its internal simulation, the common database, and state/event messages received from external entities.*
- *Each DIS entity employs Remote Entity Approximation (REA) to project a locally consistent time/space view of external entities.*
- *Simulation entities correspond closely to weapon systems and other actual equipment found in the synthetic environment.*

The last principle leads naturally to the creation of an object-oriented architecture.

Layered Architecture Model

The Strawman DIS Architecture focused on the physical implementation of networked simulation—the underpinnings of devices and networks which support the synthetic environment. The reference model has since been expanded to constitute a layered structure that gives greater expression to the synthetic environment and its algorithmic supports. The benefits of this layered model are fourfold:

- *layering simplifies module design in each layer*
- *greater emphasis on the synthetic environment supports applications and users*
- *Layer-to-layer assignment through the Architecture promotes independent and systematic exercise design*
- *Requirements trace through the Architecture layers supports enhanced Verification, Validation, and Accreditation (VV&A)*
- *Format emulation of other familiar architectural paradigms (e.g. OSI) supports acceptance and understanding of the architecture*

Figure 1 illustrates the layered reference model.

Synthetic Layer - This layer describes the synthetic environment in a hierarchical format, organized by classes and objects. It is used for descriptive purposes and provides benefit to fidelity description, VV&A, and expression of simulation requirements.

Logical Layer - This layer defines and describes entities and the underlying models of entity-to-entity communication. The format used is class and object descriptions which capture the methods and mechanisms of communication. This layer specifies entity types, allowable interactions between these types, and standard message formats

Physical Layer - This layer defines and describes the physical realization of DIS implementation. Its purpose is to describe: (1) implementation of entities, (2) implementation of interactions (including bit-wise design of PDU's and standard REA algorithms), (3) networking, (4) security, (5) interoperability, (6) strategies for VV&A, and (7) definition of "test-points" for testable, verifiable design.

Physical Layer Reference Model

The reference model for the physical layer is depicted in Figure 2. The essential components of this model are entities, cells, Cell Interface Units/Cell Adapter Units, and virtual networks. The diagram portrays three cells, one of which is shown in exploded view in order to depict its internal components.

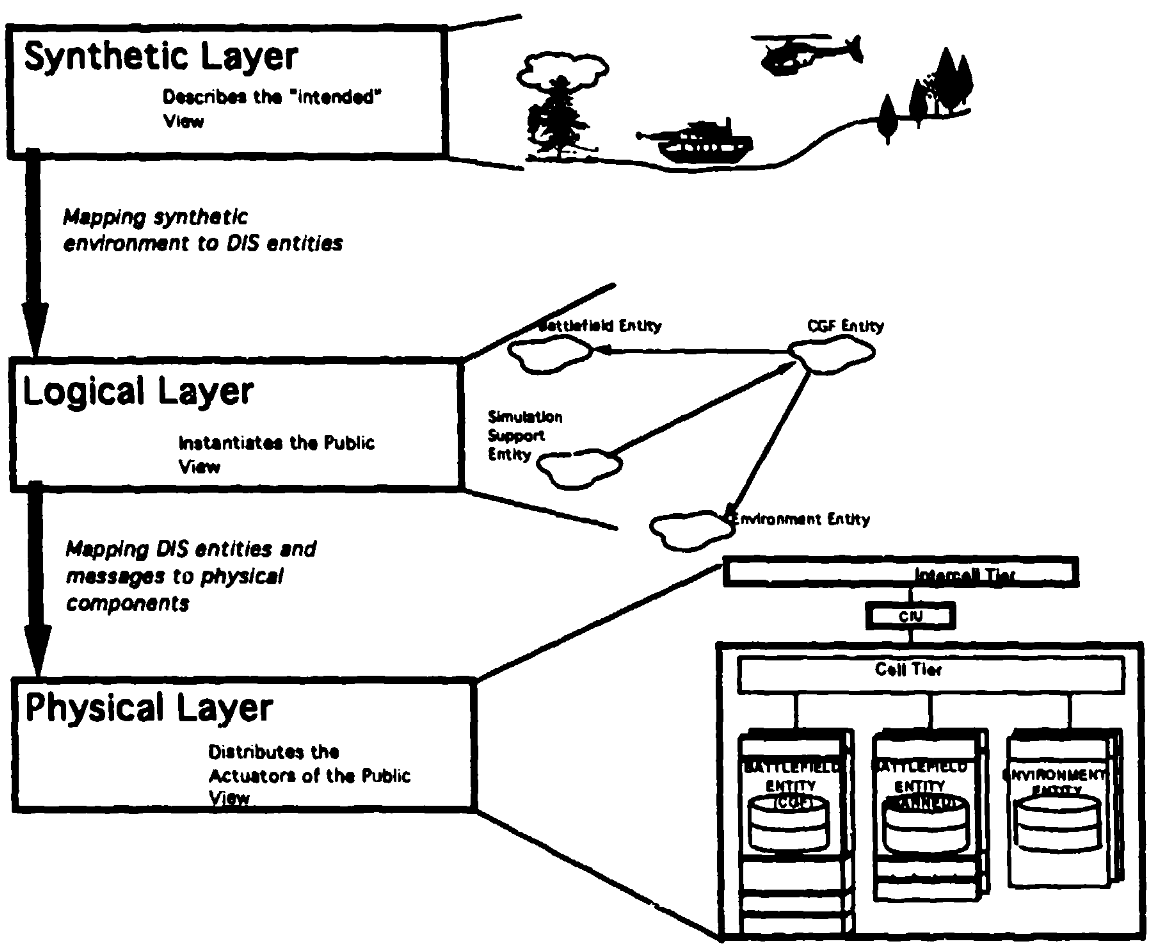


Figure1: Layered Architecture Model

Entities - Entities, as defined by the DIS Protocol are "elements of the synthetic environment that are created and controlled by a simulation application through the exchange of DIS PDU". The DIS Architecture broadens this terminology somewhat to describe simulation entities as objects that use the message-passing mechanism to interact with, *control, or monitor* the synthetic environment. The Architecture identifies three types of Simulation Entities: battlespace entities, simulation support entities, and environment entities. Battlespace and environment entities are properly "Simulation Entities" in the DIS protocol standard.

Battlespace entities correspond to actual battlespace equipment or organizations. They can be aircraft, ships, armored vehicles, dismounted infantry soldiers, guided missiles, command posts, trucks, lasers. emitters, platoon units, company units, etc. A battlespace entity incorporates a direct soldier/machine interface which emulates the soldier/machine interface associated with its real-world analog.

Simulation support entities "instrument" the simulation. They provide monitor and control, but have no analog on the actual battlefield. Examples include Plan View Display and Exercise Controller.

Environment entities corresponds to the components of the actual battlefield environment—terrain, atmosphere (haze, clouds, wind, etc.) bathysphere, sun lighting, moon lighting, and unmanned objects in the environment. They have no direct soldier/machine interface.

Asset vs. Entity - The Architecture uses the term "asset" to distinguish a simulation resource from its network presence. Entities, as described above, have network presence. As such, they can really be said to belong to all three layers of the Architecture. An asset is a physical resource (e.g. computer, cable, human operator, database) that supports an entity. To promote "plugable" interoperability, assets must be managed by the Architecture as well as entities.

Cells - Cells are collections of entities. Cells come in two varieties: Standard and Non-Standard. A Standard Cell contains only entities whose public parts conform to the DIS Architecture. A Non-Standard Cell does not. Non-Standard Cells may consist of non-DIS simulators, instrumented live vehicles on a training range, or analytic combat models.

All entities which share common databases and models and which are therefore "compatible" in the synthetic environment, belong in a common cell. DIS compliance, in the case of a Standard Cell, means that constituent entities communicate via the DIS protocol, and they draw their data from DIS-compliant Cell databases.

Figure 2 also describes the structure of the DIS Cell Database. This structure is known as the Common Database Standard (CDB) and is part of the Logical Layer. The DIS CDB consists of three component databases: Simworld Database, Session Database, and Review Database. The Simworld Database defines the underlying models of the synthetic environment (e.g. remote entity approximation algorithms, atmospheric models, terrain models, weapons and weapon effects models, rendering algorithms) and has a key role in supporting interoperability by ensuring commonalty of functionality among models and algorithms distributed across the network.

Virtual Networks - The virtual network connects entities for the instantiation of the synthetic environment. It has a two-tier structure. The cell-tier connects entities within a cell, the inter-cell tier connects cells. The Architecture specifies that the inter-cell tier must be DIS compliant. Linking the two tiers are devices known as Cell Interface Units (CIUs) and Cell Adapter Units (CAUs). CIU's connect Standard Cells to the upper

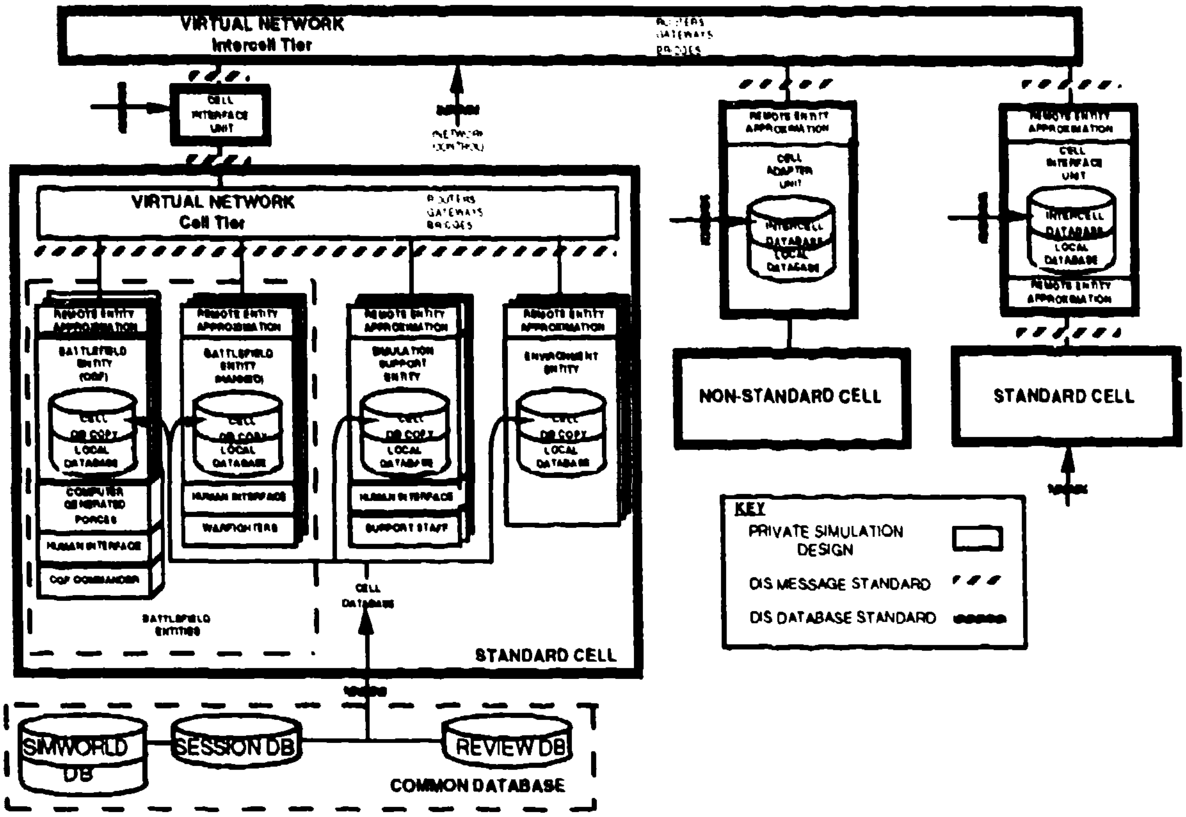


Figure 2: Reference Model for the Physical Layer

tier. CAU's connect (or adapt) Non-Standard cells to the upper tier.

Interoperability "Classes" and "Domains" — Two concepts in the Physical Layer directly address interoperability. The first, the Simworld Database, *identifies* "interoperability classes". The second, the cell, *contains* "interoperability domains". An "interoperability class" refers to a set of simulator (asset) characteristics which are sufficiently compatible in terms of algorithms, models, fidelity, resolution, database, security, throughput, physical connectivity, etc. to *validly* interoperate in most exercises. A cell contains a group of simulators, whose characteristics are all drawn from the same interoperability class, which are linked together to *meaningfully* interoperate. A cell is real assets linked together to support real exercises, hence an "interoperability domain".

Interactions between cells occur via CIUs and the CAUs. So here too the architecture addresses interoperability.

SITE DESCRIPTIONS

To sharpen its focus on the key issues of interoperability, we have concentrated Architecture extended development onto a BDS-D subsystem which highlights some of the relevant problems of interoperability. We call this subsystem the Architecture Design-Focus System (ADFS). A block diagram of the ADFS is presented in Figure 3.

The ADFS consists of three different cells (two existing, one proposed) that are being networked together to support joint exercises. Two cells are already connected: the Aviation Testbed facility (AVTB) at Ft. Rucker, AL, and the Crew Station Research

and Development Facility (CSRDF) at NASA Ames Research Center in Mountain View, CA. The third cell (proposed) will be hosted at the AVTB site and will consist of upgraded flight simulators. In this paper, this new cell will be referred to as the "Level II Cell"— "Level II" being the designation applied to simulators in this cell, to distinguish their greater capability over the older "Level I" simulators of the AVTB.

Aviation Testbed (AVTB) Cell

Of the three cells, the Aviation Testbed (AVTB) has the most simulators in terms of both numbers and types. It represents mid-1980s technology, and was a key prototype system for DIS technology.

Connectivity - Simulators in the AVTB complex are linked via a standard 10 Mbps Ethernet. They communicate via the SIMNET (SIMulator NETworking) message protocol. The Ethernet is connected to a single long haul network by a gateway computer.

Simulators - The AVTB comprises several kinds of man-in-the-loop simulators. Among them are: generic rotary-wing aircraft (RWA), generic fixed-wing aircraft (FWA), M1 Tanks, M2/M3 Infantry Fighting Vehicles, and generic air defense devices (GADD's).

Image Generators (IG's) - For the aircraft simulators, visual imagery is generated through eight TV monitors by a dedicated IG. The IG outputs nine channels of video—eight low-resolution channels for out-the-window (OTW) visuals, and a single high-resolution channel for sensor simulation. The total field of view available is 125 degrees horizontal by 30 degrees vertical. The OTW views are vertically slewable and update in real time at a 15 Hz frame rate. The sensor views replicate day TV (DTV) and forward looking infrared (FLIR), and have various fields of view that are selectable by the CPO or CPG.

For the ground vehicle simulators, a dedicated IG generates eight low-resolution channels—seven are for vision blocks and one is for the gunner's primary sight (GPS).

Both types of IG systems are optimized for the display of moving models, a capability which figures prominently in tactical scenario simulation.

Tactical Environment - A system known as Semi-Automated Forces (SAFOR) represents enemy and auxiliary forces in the synthetic environment. SAFOR is a system of workstation-controlled computer generated forces that interact with manned simulators on the battlefield. Each workstation is capable of creating a battalion-size force. The purpose of SAFOR is to provide a larger battlefield context without being operator-intensive. SAFOR units are commanded by the Workstation operator who can execute pre-planned scenarios or create responses to evolving battlefield situations.

Level II Cell

The Level II Cell will consist of the latest-generation, modular, re-configurable man-in-the-loop simulators. These devices will be configured as generic RWA's, but the design will easily extend to other vehicle types. Plans call for a total of eight Level II simulators to be placed at this cell.

Connectivity - The Level II simulators will be connected by a private network, separate from the AVTB Ethernet. This network will utilize a state-of-the-art Fiber Distributed Data Interface (FDDI). Message communications over the FDDI link will conform to the DIS protocol.

Simulators - The Level II simulators will consist of state-of-the-art modular, reconfigurable cockpits. They will comply with the USAF-developed MODSIM architecture. Host computing devices will consist of distributed microprocessors running a real-time UNIX operating system.

Image Generators - Two candidate systems are under consideration for use with the level II simulators. Both systems

represent the best of present-day, mid-priced IG technology. Common performance features between the two systems are high-resolution displays, fast update rates, and display of multiple moving models.

The first candidate system consists of three IG subsystems linked together by a high-bandwidth network. Through this network, the three systems share a common interface to the host computer for viewpoint and moving model control. Each of the IGs has access to the common terrain database making the three subsystems effectively operate as one IG. Two of the IGs are dedicated to OTW visuals. These subsystems drive three video channels each. The remaining subsystem drives two video channels and is dedicated to sensor simulation.

The second candidate system was under development at the time of writing of this paper and design details are still sketchy.

Tactical Environment - The Level II cell will have a newer-generation SAFOR system known as ModSAF (for Modular SAFOR). ModSAF will have greater capability than the existing SAFOR system and will be able to simulate much of the sensor and EW phenomenology of the modern battlefield.

CSRDF Cell

The Crew Station Research and Development Facility (CSRDF) brings the highest fidelity of helicopter simulation to the ADFS. CSRDF is an engineering simulation designed to support helicopter development. It therefore significantly differs from the tactical training simulations mentioned above.

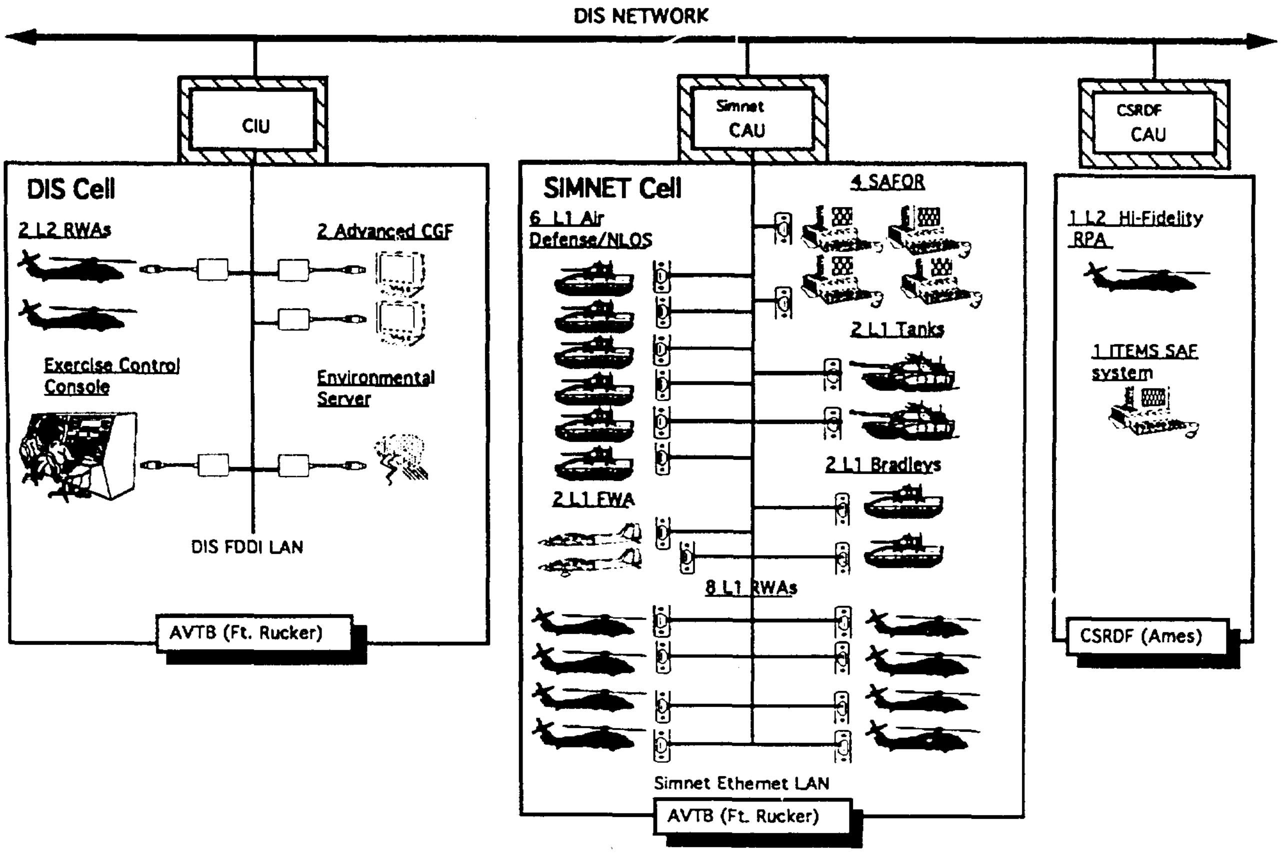


Figure 3: System Block Diagram (with Cells noted)

Connectivity - The CSRDF facility is connected, via a long-haul gateway, with the Ft. Rucker AVTB facility.

Simulators - The CSRDF cell consists of one simulator configured as a two-seat cab on a fixed platform. It was designed originally to support evaluation of a 2-vs-1 crewmember question for the Army's LHX program, and was subsequently used to train LHX simulator assessment pilots prior to their visits to contractor sites. Since then, the CSRDF has been used for rotorcraft human factors research studies.

Image Generator - The CSRDF IG represents mid-1980's high-end technology. This four channel system is capable of displaying several modes: color OTW, FLIR, and DTV. A fiber optic helmet mounted display (FOHMD) is used to display visual information to the pilot. Of the four channels, one is used for each of the left and right background displays on the FOHMD, one for the high resolution inset display, and the remaining for the Automatic Target Recognition System.

Tactical Environment - The Interactive Tactical Environment Management System (ITEMS) is a software package being developed for CSRDF. ITEMS will be used for creation, control, and execution of the tactical scenarios with which the simulator crew will interact.

ITEMS will create and control all aspects of the tactical scenario: players (air and ground), player tactics, intelligent companions, adversaries, and gaming area weather. The capability of each player will be modeled to include maneuvering, active and passive sensors, weapons, signature and communications. Detailed modeling of guided, ballistic and static weapons will also be included.

INTEROPERABILITY CHALLENGES

Significant differences in simulation capability exist between the three ADFS cells. Integration of these systems, so that they can interoperate in meaningful training and research exercises, is contingent on solving problems that arise due of these differences.

Image Generator and Display System

Unquestionably, a major impact on interoperability is the differences in visual feedback received by trainees from their individual IG and display system. Due to the complexity of this subject, it is impractical to describe each of the four systems in detail. Instead, Table 1 summarizes those differences that can impede interoperability. Please note that capabilities described for each of the four systems are as procured for this application. System vendors may support additional capability through advanced products and enhanced options to the products described.

IG systems simulate both the OTW view and the sensor suite available to vehicle crews—Direct-View Optics (DVO), Day TV (DTV), Low-Light Level TV (LLLTV), and Forward-Looking Infrared (FLIR). In the paragraphs below, we offer additional explanation for the entries in the tables.

Update Rate - This characteristic describes how frequently the visual scene is updated or "re-painted" on the displays. Faster update rates lead to less scene jitter and more realistic, smooth motion.

Diurnal Effects Simulation - IG systems may simulate the changes in color and lighting that occur in the visual scene with the passing of the day (dawn, day dusk, and night).

Visual Rendering Range - This characteristic describes the simulated viewing range at which an IG system can render visual objects. The comparison in Table 1 presents "best-case" rendering range, ignoring database density issues.

Moving Models - Because BDS-D is a tactical simulation, the number and complexity of moving models displayable is critical. Articulated components of moving models, such as rotating turrets, lend significant cues to warfighters. Moving

	L II (#1)	L II (#2)	AVTB	CSRDF
OTW Update Rate	30 Hz	30 Hz	15 Hz	60 Hz
Sensor Update Rate	60 Hz	60 Hz	15 Hz	60 Hz
Display Type	Dome	Dome	CRTs	FOHMD
OTW Resolution	4.17 arc min.	2.81 arc min.	4.13 arc min.	3.56 arc min.
Sensor Resolution	5.62 arc min.	5.62 arc min.	4.68 arc min.	3.56 arc min.
Diurnal Effects	Yes	Yes	No	Yes
Multiple Sensors	Yes	Yes	No	Yes
OTW Range	10 km	10 km	3.5 km	? ?
Sensor Range	15 km	15 km	7 km	? ?
Atmos. Attenuation	Yes	Yes	No	Yes
Total 6 DOF Models	100	74	64	16
Weapon Effects (3 DOF)	No Additional	40	128	32
Rockets/Tracers (3 DOF)	40	30	No additional	No additional
Polygons/Sec. - Total	396,000	360,000	210,000	240,000
Transport Delay - OTW	100 ms	100 ms	167 ms	78 ms

Table 1: IG and Display System Comparison

model display methods differ significantly between IG's and can be difficult to quantify.

Scene Density (System Output Performance) - The polygon display output of an IG determines the scene density that can be rendered. From a fidelity aspect, high density allows a closer representation of the detail of real visual scenes, providing the key motion and position cues required when flying close to the earth's surface. In applications like our aviation-oriented testbed, high scene density is critical.

Scene density is measured in terms of number of polygons displayed per unit time.

Resolution - Resolution is defined as the angle which is subtended by a pair of adjacent television raster lines on the image plane when measured from the design eye. The subtended angle is expressed in Arc Minutes of resolution.

Other IG Effects (Database and Load Management)- When the density of polygons in the database causes rendering overload, actions taken by the load management model can introduce unpredictable visual effects which can unbalance the fair fight. For example, IG "A" can be experiencing overload while IG "B" is not. Under normal loads each IG should render models at identical ranges, but in this example, IG "A" will pull in the rendering ranges of models, giving "B" a range advantage for target detection.

Occurrences of situations like this need to be kept to a minimum by utilizing good data base analysis and design practices.

Intervisibility

Pairwise-discrepant intervisibility between IG systems occurs when simulator A has clear line-of-sight to simulator B ,B has obstructed line-of-sight to A, and this discrepancy is due to an IG or database anomaly, not because of a reproduction of real-world features.

Miscorrelation between databases is one cause of this problem. Simulator A may be intending to hide behind a rock or a house that appears in its IG database. If the obstacle is positionally miscorrelated in B's database, the result may be that B can see A, A cannot see B (because of the obstacle), and A believes he is hidden from B. These problems can only be solved by a rigorous program of correlation.

IG-caused discrepant intervisibility is more problematic. Anomalies occur because of IG load management schemes. When confronted with more visual density than it can instantaneously process (too many polygons and too many moving models), an IG must adopt scene management techniques. Available strategies may include: not rendering distant terrain, dropping the level-of-detail of rendered models, and less frequent update of moving models. Any one of these strategies can cause discrepant intervisibility between scenario players.

Tactical Communications

Radio signal attenuation and distortion must be modeled so as to achieve uniform phenomenology in the synthetic environment. This problem is akin to the intervisibility problem.

The solution is to adapt a single-point "radio server" on the network that arbitrates all connection decisions by modeling attenuation and distortion.

Network Capabilities

Figure 3 portrays the network connections between the three cells of our design-focus system. The network configuration is highly mixed—local area network connections (LANs) of differing protocol, long-haul connections, and gateways. Differences in operational bandwidth between components of this network may impact interoperability.

Traffic on this network will consist mostly of the messages which describe entity positions and events. As additional entities join an exercise, network utilization proportionately increases. As network links begin to saturate, simulation entities begin to experience late message delivery and outright lost messages. Under extreme saturation, an affected link may fail entirely, dropping its dependent entities from the exercise.

Task Performance Fidelity

Differences in task performance fidelity may also affect interoperability via differences in warfighter workload. As the workload of the basic piloting tasks (flying, navigation, target recognition, acquisition, etc.) varies among the systems, simulation and training outcomes may vary with respect to what would occur in the real-world.

Tactical Environment

Differences in capacity and thinking ability of computer generated forces may also affect interoperability in terms of the fair fight.

INTEROPERABILITY AND ARCHITECTURE

Interoperability Types

The context of the Architecture, and its layered structure, allows us to clarify notions about interoperability.

A precursor to interoperability must be correct system and hardware interaction—the medium of communication between systems. These connections are documented and accounted for under the Physical Layer.

The initial threshold of interoperability is crossed by the implementation of the DIS Standards—the message protocol and supporting database standards. This implementation is described in the logical layer. This condition of meeting minimum interoperability requirements, we shall call *weak interoperability*.

Weak interoperability is not satisfactory for most simulation applications. We understand that interoperability can only be measured in the context of the synthetic environment. Starting with weak interoperability, one can proceed in two somewhat different directions in describing greater positions of interoperability. The first is a quantitative reckoning of the number of simulated functions that can interoperate. The second is a qualitative reckoning of the degree of interoperation versus real-world functionality. Complete quantitative interoperability we shall call *strict interoperability*. Complete qualitative interoperability, (an unattainable ideal) we shall call *strong interoperability*.

Strict interoperability means that the two systems interact appropriately in all modeled functions. For example, two different high-fidelity simulators may interact with each other in many areas of simulation (e.g.. sensors, visual appearance, navigation) yet not be strictly interoperable because of their implementation of different flight models levies different workload requirements on the subject pilots. Yet two different simple simulators, whose models do not account for much of the detail of the real world, may be strictly interoperable because of the inclusiveness of their interactions.

For systems that interoperate over many kinds of real-world functions, we shall say that they exhibit *strong interoperability*. There is no theoretical maximum to strong interoperability because of the inexhaustible amount of detail in the real world and our limited ability to represent it.

Two points can be made on this distinction.

(1) Strict interoperability and strong interoperability are not purely orthogonal concepts. For systems to be strongly interoperable, they must first share a high degree of strict interoperability.

(2) Strong interoperability is the most useful measure of interoperability for purposes of VV&A.

Figure 4 illustrates how the ADFS cells relate to these concepts.

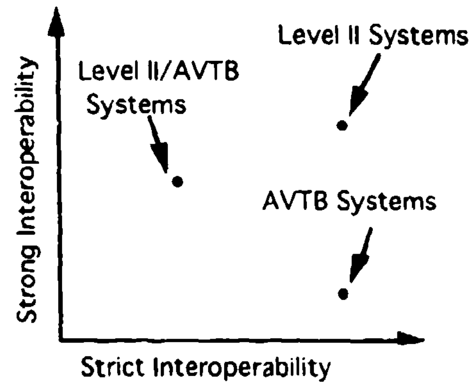


Figure 4: "Interoperability Space"

The "Fair Fight"

Interoperability directly affects the *fair fight*. When simulation is used for training or experimentation, the user intends to substitute simulation outcomes as surrogates for real-world outcomes. Hence the user tries to correlate simulation outcomes with outcomes in the real world. The term "fair fight" indicates that a strong correlation exists. The fair fight is related to strong interoperability in that it is evaluated by comparison to the real world.

There are two causes for the reduction of "fairness" in the fair fight during a simulation exercise: loss of time-space correlation and differing granularity in simulation fidelities.

Loss of time-space correlation can stem from miscorrelation among databases, "distribution effects", and simulator-specific differences. Miscorrelation of databases has already been discussed. Distribution effects are network-related anomalies such as latency, dropped packets, and out-of-sequence deliveries. These items can lead

to a distortion of perceived time between two different systems in the exercise. Simulator-specific differences encompass database density and concomitant load management schemes.

Granularity in relative simulation fidelities can lead to differing workloads for pilots in the simulation.

CONCLUSIONS FOR THE ARCHITECTURE

In late 1992 and early 1993, an effort was undertaken to link two cells of the ADFS—the CSRDF cell and the AVTB cell. Initial requirements for this integration were not overly stringent. The key technical problems encountered involved: obtaining, converting, and matching terrain databases; and translation between two partially-disjoint message protocols.

Our experience with this integration effort, and with the design of the ADFS in general, generates valuable feedback for the future shape of the architecture in support of interoperability.

Validation of Need for Architecture Components

The most immediate results for the Architecture are the validation of the need for certain design components identified in the Physical Layer.

CAU—Because the AVTB utilizes the non-DIS-compliant SIMNET protocol, the first step toward the CSRDF-AVTB linkage was the development of a CAU to adapt the SIMNET protocol to the DIS protocol used on the long-haul network. A principal lesson learned is that the CAU must be of maximally high performance to sustain an exercise. Impediments to system throughput include the mathematically-intensive positional coordinate conversion routines. For this reason, project engineers experimented with substituting these routines with a polynomial-based mapping to promote greater speed.

Common Database (CDB)—The "handcrafting" of the terrain databases that had to occur to support interoperability shows the need for a strong CDB standard.

Environment Entities— These single-point arbiters of line-of-sight and radio connectivity are key to preserving the simulation illusion.

Reality of Exercise Design

The CSRDF-AVTB linkage relied on an obvious solution to the interoperability challenge—utilizing the lowest common capability between linked cells. This solution, "exercise design" (or downgrading), can be achieved by restricting exercise parameters to just those capabilities which can be systemically supported. Realistically, most future applications of DIS will have to rely on exercise downgrading to support interoperability.

The down side of this solution is obvious. Consistent application would require that, as long as there is at least one less-capable system in the exercise, all higher-capability simulators must play down to it. This strategy would neutralize the current investment in simulation equipment.

However, given the current situation, the Architecture should robustly support exercise downgrading. Current support consists of the means to capture and describe the reduced exercise capabilities through the class descriptions of the Synthetic Layer and the structures of the SIMWORLD Database. This descriptive capability promotes not only up-front exercise design, but also post-design validation and accreditation.

Given the importance of this area, developing the Architecture toward further support of exercise design would be beneficial.

Assert Interoperability at the Synthetic Layer

In discussion of Interoperability, attention must be focused on the phenomenology of the synthetic

environment. Compliance to a standard message protocol is merely a precursor to interoperability. Meaningful interoperability can be assessed by use of the Synthetic Layer of the Architecture. To do so, one first identifies the intersection of the relative capabilities of the distinct simulation synthetic environments (as described by the Synthetic Layer class hierarchy). Then one identifies which parts of the intersection are implemented in the resulting combined synthetic environment.

Limitations of "Non-Invasive" Architecture

A "non-invasive" architecture (one that attempts to minimize required changes on legacy assets to have them play in DIS) will have limitations when it comes to supporting interoperability. Mere implementation of the message protocol is not enough to support full-bodied interoperability. One must be willing to modify legacy assets in order to support the degree of interoperability desired. A requirement for the Architecture, that stems from this realization, is that the Architecture must provide design guidance for such reconfiguration.

Floating Public-Private Boundaries

The architectural boundaries between the public and private parts of assets must not be considered a fixed line, but must be viewed as "floating" based on the degree of interoperability desired. The Architecture must support definition of this boundary.

ACKNOWLEDGMENTS

This paper has been supported by the U.S. Army's Simulation, Training, and Instrumentation Command (STRICOM) under the ADST contract.

The author would like to acknowledge the following individuals for their helpful input during the preparation of this paper: Bob Edwards, Steve McCarter, Dr. Richard Sherman, and Chuck Kuczaj.

REFERENCES

[1] Distributed Interactive Simulation Architecture Description Document. By Loral Systems Co. Orlando, FL: Simulation, Training, and Instrumentation Command, 1993.

[2] Distributed Interactive Simulation Common Database Standard. By Loral Systems Co. Orlando, FL: Simulation, Training, and Instrumentation Command, 1993.

[3] Kuczaj, Chuck. "ADST CSRDF Protocol Translator Integration & Test, Ft. Rucker Alabama, February 22, 1993—March 11, 1993." Trip Report, Loral Western Development Labs, 1993.

[4] Hsu, J., Greenfield, B. "The Fiber Distributed Data Interface Emerges As 100 Mpbs LAN Solution." Computer Technology Review. November, 1990.

[5] Rotorcraft Pilots Associate (RPA) Advanced Technology Transition Demonstration (ATTD), Appendix J, Crew Station Research and Development Facility (CSRDF). Author unknown. RPA bidder's conference handout, 1992.

[6] Sherman, Richard. "CSRDF Lessons Learned." Private Communication, Loral Western Development Labs, 1993.

[7] Standard for Information Technology—Protocols for Distributed Interactive Simulation Applications, Version 2.0. By The Institute for Simulation and Training, Orlando, FL. 1993.

[8] System Definition Document. By Loral Systems Company. Orlando FL: Simulation, Training, and Instrumentation Command, 1992.

DISTRIBUTED^3 INTERACTIVE SIMULATION AT ACETEF

Phil Landweer
BDM Federal, Inc.
Albuquerque, New Mexico

ABSTRACT

The Air Combat Environment Test and Evaluation Facility (ACETEF) used distributed interactive simulation and computer generated forces in support of two T&E programs during the summer of 1992. The first of these involved testing the effectiveness of an Electronic Countermeasures (ECM) system used to protect tactical aircraft during a strike against a ground- and surface-based Integrated Air Defense System (IADS). The scenario included a full Navy strike package with 16 aircraft and a robust threat with over 30 units. Three manned simulators, three actual hardware systems, an RF environment stimulator, and a digital simulation (the Simulated Warfare Environment Generator, or SWEG) were integrated in real-time to provide a full-up, closed-loop combat environment to the ECM system under test. The second test investigated the relative contributions of various E-2C communication systems to the effectiveness of a carrier-launched tactical strike against a Surface Attack Group (SAG). The tactics used during the strike were specified by a Naval aviator, with the CGF models adjusted to represent these specific behaviors. As part of the test planning process, several hundred stand-alone SWEG runs were made to predict what would happen during the actual test. Again, a variety of simulators, hardware, stimulators, and digital simulations were integrated in a real-time environment using SWEG. All incidents of interest were captured by SWEG as the test scenarios progressed, with a post-processor used to quantify the various measures of effectiveness (MOEs) identified by the test analysts.

ABOUT THE AUTHOR

Phil Landweer works for BDM Federal, Inc. as the manager for Advanced Simulation. He has used digital modeling and distributed simulation for concept exploration, requirements analysis, pre-test analysis, test planning, test and evaluation, and tactics development to support a variety of government and industry organizations. His specific areas of interest include object-oriented simulation, functional modeling, and integrated computer graphics. Prior to joining BDM, he was an analyst at the Air Force Operational Test and Evaluation Center. Mr. Landweer holds an MS in Applied Mathematics from Carnegin-Mellon University, and MBA from New Mexico Highlands University, and a BS in Computer Science, Operations Research, and Mathematics from the US Air Force Academy.

DISTRIBUTED^3 INTERACTIVE SIMULATION AT ACETEF

Phil Landweer
BDM Federal, Inc.
Albuquerque, New Mexico

OVERVIEW OF ACETEF

The Air Combat Environment Test and Evaluation Facility (ACETEF) is a collection of laboratories and their associated simulators, hardware, software, and simulations which replicates a combat environment for System(s) Under Test (SUT). Several laboratories make up ACETEF proper (see figure 1).

The *Anechoic Chamber* at the center of the diagram is a non-reflective, isolated enclosure for creation of a free space test environment for any SUT located in the chamber. The *Manned Flight Simulator* (MFS in the diagram) provides man-in-the-loop control for an aircraft, perhaps located in the anechoic chamber, with a realistic cockpit environment including visual and motion control. The *Advanced Flight Simulator* (AFS) will enhance the current capabilities of the MFS. The *Tactical Avionics and Software Test and Evaluation Facility* (TASTEF) uses the 1553A Multiplex Bus and Naval Tactical Data System (NTDS) to simulate the airborne environment for testing of avionics and associated software. The *Electronic Warfare Integrated Systems Test Laboratory* (EWISTL) provides a radio frequency (RF) simulation and stimulation of threat radar signals to electronic warfare (EW) systems on the SUT. It includes the Enhanced Threat Electronic Warfare Environment Stimulator (ETEWES), which generates the RF signals within the anechoic chamber. The *Closed Loop* (CL) facility uses simulations of threat weapon systems in a closed-loop fashion which are fully responsive to counter-measures taken by the SUT. The *Electromagnetic Environment Generation System* (EMEGS) provides RF simulation and stimulation of the SUT at high power levels such as those found on the flight deck of an aircraft carrier. EMEGS is currently limited in the number of concurrent platforms it can support. The *Electromagnetic Environment Effects Test Laboratory* (E3TL) will increase the number of platforms supportable, and bring other improvements over EMEGS. The *Offensive Sensors Laboratory* (OSL) will provide RF, infrared (IR), ultraviolet (UV), and laser simulation/stimulation of offensive avionics and weapons systems for the SUT. It will also simulate/stimulate other portions of the electromagnetic spectrum. The *Communications, Navigation, and Identification Laboratory* (CNIL) is currently being developed, and provides RF and digital friendly and threat stimulation of CNI systems for SUT in concert with simulation/stimulation of other simulated player CNI systems. The *Aircrew Systems Evaluation Facility* (ASEF) provides man-in-the-loop player control during ACETEF tests for the purpose of evaluating man-machine interfaces and human factor issues. Finally, the outer ring of the figure represents the *Operational Control Center* (OCC), which provides simulation and control for multi-player/multi-laboratory test using the Simulated Warfare Environment Generator (SWEG) and up to 15 mini-crewstations which will provide Red/Blue man-in-the-loop capability for additional units. These will include air, ground, and surface combatants, as well as Command and Control nodes and White Control Team workstations.

In addition to the ACETEF-proper facilities shown in the diagram, other laboratories and assets can be integrated. The Real-time Electromagnetic Digitally Controlled Analyzer and Processor (REDCAP) was successfully integrated into an ACETEF scenario in 1990, and the E-2C Simulator Laboratory (ESL) can also be linked in to a test.

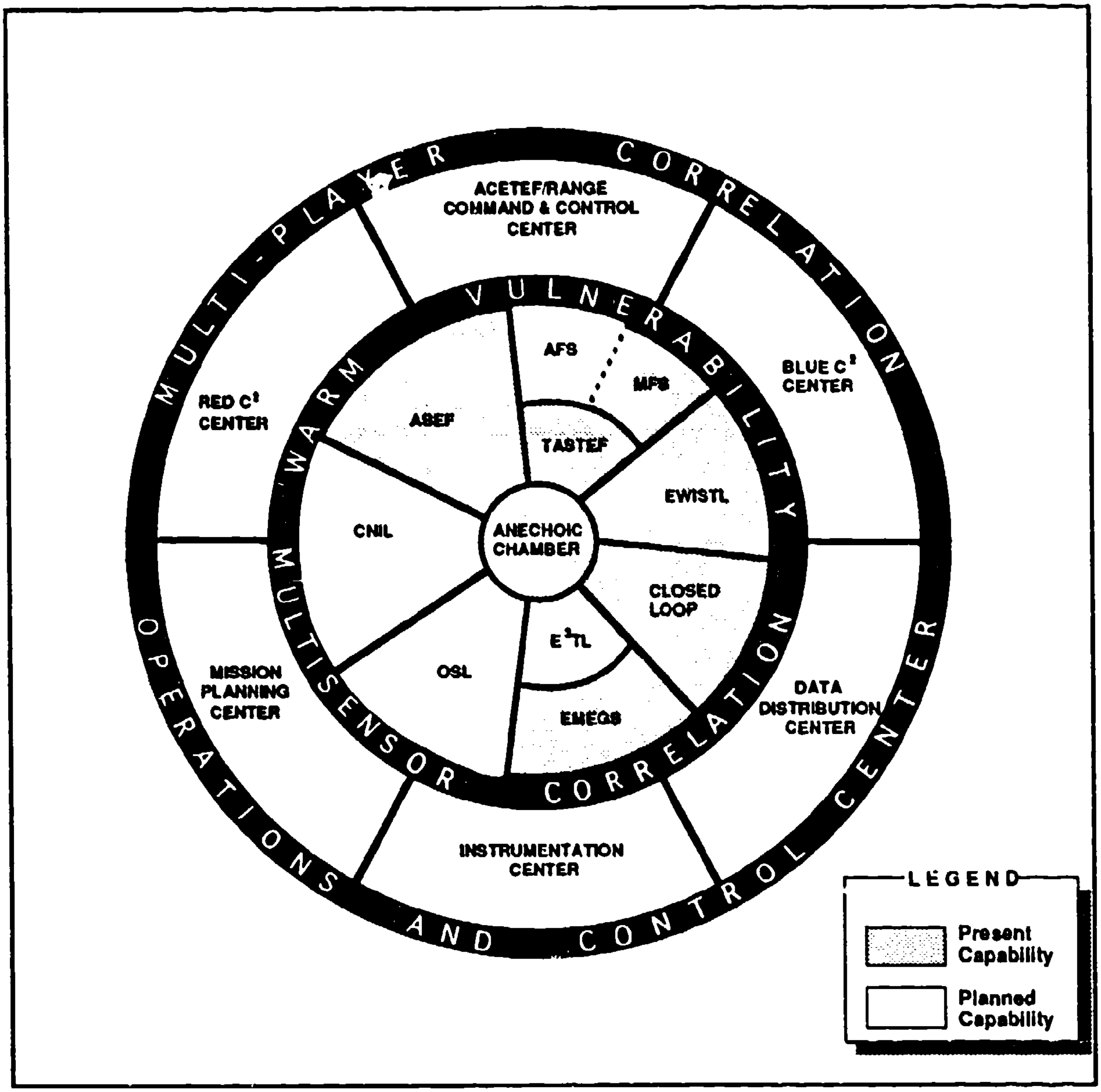


Figure 1. Present and Planned Composition of ACETEF

ACETEF ARCHITECTURE USED DURING TESTS

During June of 1992, a demonstration test to the Joint Aeronautical Commander's (JAC) Group was run. It involved testing the effectiveness of an Electronic Countermeasures (ECM) system used to protect tactical aircraft during a strike against a ground- and surface-based Integrated Air Defense System (IADS). The scenario included a full Navy strike package with 16 aircraft and a robust threat with over 30 units. An F/A-18 located within the anechoic chamber was the SUT for this test. Actual hardware systems on-board the F/A-18 were stimulated by ACETEF assets to put the SUT into a realistic combat environment, with the RF environment provided by ETEWES. Manned simulators within the MFS, CL, and ESL facilities were used to increase the realism of the test. SWEG performed the roles of integrating these assets into an overall scenario. as well as modeling all unit functionalities not represented elsewhere. Thus, assets within the anechoic chamber, MFS, EWISTL, CL, CNIL, OCC, and ESL were integrated together to replicate a

combat environment for the F/A-18 SUT. Live, virtual, and constructive simulations were interacting with each other during the scenario.

In September and October of 1992, a similar test at ACETEF was conducted. It investigated the contribution of various E-2C Hawkeye communications configurations to a carrier-launched tactical air strike conducted against a Surface Attack Group (SAG) in a war-at-sea scenario. A configuration of ACETEF assets similar to the June test was used.

FUNCTIONAL RESPONSIBILITIES OF ASSETS

The June scenario was specifically comprised of the F/A-18 SUT and other F/A-18s providing suppression of enemy air defense (SEAD) for an air strike conducted by A-6E aircraft. EA-6B aircraft provided jamming support in an attempt to further limit the capabilities of the enemy air defense system. An E-2C aircraft provided command and control for F-14 interceptors on combat air patrol (CAP). Also on the BLUE side were tomahawk cruise missiles, which were launched against an enemy airfield.

On the threat side, the IADS was under operational control of a command post. The command post had several surface-to-air missile (SAM) systems at its disposal, along with MiG aircraft launchable from an airbase. Several early warning/ground-controlled intercept (EW/GCI) radars were networked into the command post. A Kirov-class guided missile cruiser was also in the scenario. The ship had a variety of SAM and anti-aircraft gun systems on board.

The functionalities of the units within the scenario were assigned to various simulator, hardware, and stimulator assets (see figure 2). The chart shows how the functions were partitioned across the ACETEF facilities. Down the left side of the figure are the functions which were present in the scenario. Movement allows the simulated players to change position within the scenario. *Movement* includes the flight of an aircraft, launching of missiles, and a ship moving across the sea. *Sensing* is used for players to non-cooperatively gather information on others. Radars, infrared devices, warning receivers, and visual acquisition of targets are all modeled via sensing. *Comm* is short for communications, which allows the cooperative exchange of information between players. This includes one- and two-way radio transmission, data links, and normal talking between individuals. *Jamming* is used to interfere with other players' sensing and communications functions, as well as changing the effectiveness of weapons. It incorporates such non-lethal engagement tactics as the use of ECM, chaff, flares, and smoke. *Asg/Eng* stands for assignments and engagements. These functions represent the decision to use subordinates (for assignments) and weapons (for engagements) against a potential enemy. *Firing* is the act of using one or more rounds of ordnance against a target. It is used to put bombs on target, fire missiles, and detonate warheads, for example. *Lethality* is the calculation of any damage effects, such as probability of kill (p_k), given that ordnance has arrived at a target. Finally, Flyout represents the function of calculating where ordnance will go once it has been fired. Each row of the figure shows which ACETEF asset had responsibility for that function, or "---" if the function was not present for a particular player type within the scenario.

Across the top of figure 2 are the player types which made up the BLUE forces during the test. The SUT was an F/A-18 during this particular test. Continuing across the top are F/A-18, A-6E, EA-6B, E-2C, and F-14 aircraft. Tom. stands for the Tomahawk cruise missiles which were launched during the scenario.

The functions of the SUT were distributed across several ACETEF assets. The column headed with "SUT" shows the particular distribution used. The Manned Flight Simulator provided the movement of the SUT, with a Naval aviator flying within a full-domed F/A-18 simulator. As the simulator was flown around, its position was updated within SWEG, which in turn allowed all of the other ACETEF assets to properly place it within the scenario as necessary. The sensing functions of the SUT were done by three assets. The visual sighting of other players was done by the pilot within the MFS, with the various platforms projected at the proper position, aspect, and apparent size on the simulator's

	SUT	F/A-18	A-6E	EA-6B	E-2C	F-14	Tom.
Movement	MFS	SWEG	SWEG	SWEG	SWEG	SWEG	SWEG
Sensing	MFS SWEG H/W	SWEG	SWEG	SWEG	SWEG	—	SWEG
Comm	CNI Lab	—	—	—	CNI Lab SWEG	SWEG	—
Jamming	H/W SWEG	SWEG	SWEG	SWEG	—	SWEG	—
Asg/Eng	SWEG	SWEG	SWEG	—	ESL	—	SWEG
Firing	MFS	SWEG	SWEG	—	—	—	SWEG
Lethality	SWEG	SWEG	SWEG	—	—	—	SWEG
Flyout	SWEG	SWEG	SWEG	—	—	—	—

Figure 2. Distribution of player functions was spread across several ACETEF assets.

dome. SWEG modeled the ground attack radar within the SUT. Finally, an actual ALR-67 radar warning receiver within the anechoic chamber displayed any threat emitters which it detected, with this display fed back into the scenario. Communications between the SUT and the E-2C were accomplished by the Communications, Navigation, and Identification Lab, which simulated the digital Link-4A system using specialized hardware and software. Jamming effects from the SUT were done using two assets: a special piece of hardware within the anechoic chamber, and ALQ-126 effects modeled within SWEG. Thus, when enemy SAM systems or radars tried to engage the SUT, SWEG degraded their effectiveness according to ALQ-126 effectiveness data entered into the SWEG data bases. The SUT was equipped with high-speed anti-radiation missiles (HARMs), with SWEG making the decision as to which enemy radars the HARM should engage. However, the pilot within the MFS determined when to fire a HARM by squeezing the trigger on the stick. Upon doing so, SWEG flew out a HARM missile, which came into existence within the scenario after the pilot fired. The exhaust trail of the HARM would show up on the simulator dome, which was quite exciting to watch. Should the HARM reach its target, SWEG would detonate the warhead and determine any lethality effects, such as taking out enemy radars.

The next column of figure 2 shows that SWEG was responsible for all functions of the other F/A-18s in the scenario. SWEG modeled their flight paths along with maneuver logic, and modeled both their ground attack radars and ALR-67 warning receivers. Since the scenario did not require the E-2C to be in radio contact

with these other F/A-18s, no communications were present. SWEG did model their ALQ-126 ECM pods, along with their engagement logic, HARM firing decisions, as well as HARM lethality and flyout. SWEG also was responsible for these same functions for the A-6E strike aircraft. SWEG had very similar functional modeling for the A-6Es, with ground attack radars, radar warning receivers, and ECM pod effects present. Rather that firing HARMs, though, the A-6Es dropped iron bombs on the target.

SWEG was also wholly responsible for the EA-6B jammer aircraft in the scenario. However, all these aircraft had to do was fly around in a predetermined orbit, sense enemy radars with a warning receiver, and reactively focus jammer spots on various frequencies.

The fifth column shows how the E-2C was distributed across two ACETEF-proper assets and an external asset. SWEG flew the aircraft in a certain orbit, and also modeled its search radar. As track files were formed into "viable perceptions," they were electronically sent to the E-2C Simulator Lab (ESL) where they appeared on the operator scope within that simulator. A human operator would then make assignment decisions, which were made known to the SUT using the CNI Lab's Link-4A system. Communications between the E-2C and the F-14s was done by SWEG, though. As can be seen in the next column, the F-14s did not have much to do in this scenario. SWEG controlled their flight, communications with the E-2C, and jamming effectiveness of on-board ECM systems. Since it was not necessary to model the weapon systems of the F-14s for the purpose of this test, computer time was not wasted on doing so.

The last column shows that SWEG was responsible for modeling all aspects of the Tomahawk cruise missiles. Their flight paths, digital scene matching sensor, engagement of targets at the airfield, warhead detonation, and lethality effects were all simulated within SWEG. Since the Tomahawks did not dispense submunitions for this attack, the flyout entry is null.

Threat System Functional Modeling

Figure 3 shows the same functional modeling partitioning across ACETEF for the threat forces. Again, each row shows which ACETEF asset was responsible for a particular function, or if that function was not modeled for a player type. The first column is for the threat air defense Command Post (CP). The CP received reports on the attacking BLUE aircraft through its radios modeled by SWEG. The CP could assign aircraft to SAM systems, and could also launch MiG-23 interceptors. Its decisions to do so were modeled by SWEG, as indicated in the Asg/Eng row.

The next column is for the threat SAM systems. Both SAM unit commanders and fire units were present within the scenario. Their radars, communications links, assignment logic, engagement procedures, firing doctrine, missile flyout, and probability of kill (p_k) lethality were all modeled within the scenario. SWEG was responsible for all of their functions except one: when a radar was on, its mode and orientation were passed to ETEWES. That system then calculated what the apparent energy would be at the SUT based upon its instantaneous position within the scenario. RF energy with the proper aspect, power, and waveform was then injected into the anechoic chamber in order to correctly stimulate the ALR-67 warning receiver on board the F/A-18.

SWEG modeled all functions of the airbase which launched the MiGs, the early warning/ground-controlled intercept (EW/GCI) sites, and MiG-23s. The airbase was only responsible for receiving a launch order from the command post, and deciding to pass on this request to launch the airplanes. Thus, all other possible functions were not represented. The EW/GCI sites had surveillance radars, whose track files were passed along to the commander via radio communications. The MiG-23 aircraft were told to launch by the airbase over a radio net, and the MiGs would then take off and fly away from the incoming Tomahawk missiles and attacking aircraft. Thus, their weapon systems, jammers, and engagement procedures were not modeled during this test.

	CP	SAMs	Airbase	EW/GCI	Mig-23	Kirov
Movement	–	–	–	–	SWEG	–
Sensing	–	ETEWES SWEG	–	SWEG	–	CL ETEWES SWEG
Comm	SWEG	SWEG	SWEG	SWEG	SWEG	–
Jamming	–	–	–	–	–	–
Asg/Eng	SWEG	SWEG	SWEG	–	–	CL SWEG
Firing	–	SWEG	–	–	–	CL SWEG
Lethality	–	SWEG	–	–	–	CL SWEG
Flyout	–	SWEG	–	–	–	CL SWEG

Figure 3. Functional mapping of threat forces across ACETEF assets.

Finally, the Kirov cruiser was a complex simulated player, with several ACETEF assets responsible for its activity. For this test, it was decided to keep the Kirov within the harbor. Thus, its movement through the water was not necessary. The radars on-board the ship were simulated by the Closed Loop, ETEWES, and SWEG assets, depending upon the type of radar, its mode, and the particular target being sensed. Specifically, CL was used for tracking the SUT, SWEG modeled the use and perception of all other radars, and ETEWES used this information to properly stimulate the F/A-18 SUT within the anechoic chamber. Similarly, SWEG modeled the engagements, weapon firing, lethality of intercepting missiles, and missile flyout against all BLUE aircraft except the SUT. The CL facility performed most of these functions should the Kirov decide to engage and fire at the SUT, though. Notice that CL was not able to model the lethality of a missile targeted at the SUT. Because CL did not have this capability during the test, SWEG modeled the fuzing, detonation, and damage effects of a Closed Loop missile flown out against the target. CL sent SWEG the position of the simulated missile being flown in order to accomplish this.

During the E-2C test in September and October of 1992, a similar ACETEF configuration was used. Additionally, SWEG represented a variety of tactical doctrines used during a Naval tactical air strike. The tactics used during the strike were specified by a Naval aviator, with the computer-generated forces (CGF) within SWEG adjusted to represent these specific behaviors. As part of the test planning process, several hundred stand-alone SWEG runs were made to predict what would happen during the actual test. This allowed the ACETEF test assets to focus on cases of primary interest. Again, a variety of

simulators, hardware, stimulators, and digital simulations were integrated in a real-time environment using SWEG. The functional responsibilities were very similar to the JAC test in June. All incidents of interest were captured by SWEG as the test scenarios progressed, with a post-processor used to quantify the various measures of effectiveness (MOEs) identified by the test analysts.

DISTRIBUTED SIMULATION ADVANTAGES

Each test was run several times, with a variety of conditions simulated within the overall scenario. Test data was collected and analyzed to determine the effectiveness of the systems being tested. ACETEF provided a full combat environment in which to immerse the SUT. Because range time did not have to be scheduled, the tests could be run as necessary within ACETEF facilities. This type of testing is far less expensive than open-air testing using combat equipment. Another advantage of testing within ACETEF is that its shielding prevents testing from being observed by unwanted personnel.

Using human operators as part of the simulated environment increased the realism of the results. However, the flexibility of SWEG allowed the behaviors of combatants to be represented where human operators either were not available or not critical to the needs of the test. Because there were no moving combatants, though, safety was not a concern within these tests. Many times, safety concerns prevent certain activities from happening within a field test, even though such actions would certainly happen in combat. These include flying an aircraft at very low altitudes, live fire of weapons, and destruction of combat platforms.

Another advantage of the ACETEF architecture is that the user-defined configuration file specifies how the functions are to be distributed across the facilities. This allows for interactions to occur between assets on a "need to know basis," rather than simply broadcasting all events to all nodes. Thus large, robust scenarios may be run without overwhelming the physical data links which tie ACETEF together. The configuration file also lets SWEG assume any of the functions of a player within the scenario simply by changing the inputs. In this way, specific events and conditions can be presented to the SUT by removing the variation due to human intervention. More importantly, SWEG can assume the roles of other simulations, simulators, or other assets which are not available due to maintenance or scheduling issues.

Because SWEG allows any simulator or simulation to be integrated into the overall scenario, computer assets and other specialized hardware are not overburdened by performing all of the simulated activity. Instead, the functions are distributed across many facilities, as explained above in figures 2 and 3. The advantages of incorporating actual hardware into a test environment are obvious. In addition, using existing simulations which are accepted by the test community minimizes the amount of verification, validation, and accreditation (VV&A) that must be done. In fact, SWEG has recently been modified to incorporate the ESAMS models for surface-to-air missile engagements, and TAC BRAWLER for air-to-air combat simulation.

CONCLUSIONS

The tests run at ACETEF during the summer and fall of 1992 demonstrated the utility of using distributed simulation for test and evaluation purposes. This technology is a very cost-effective way to test modern weapon systems under realistic conditions.

CONNECTIVITY FOR THE HIGHLY DYNAMIC VEHICLES IN A REAL AND SYNTHETIC ENVIRONMENT (HYDY) PROJECT

Kevin E. Boner
Naval Command, Control and Ocean Surveillance Center, RDT&E Division
San Diego, California

INTRODUCTION

The Advanced Research Projects Agency (ARPA) has created the Intelligent Gateway/Scaleable Simulation (IGSS) project to perform research into problems associated with large-scale simulations, combining both real and simulated units on Local Area Networks (LANs) and Wide Area Networks (WANs). The program's goal is to achieve seamless simulation by providing worldwide access to multilayer simultaneous, realtime, very large virtual warfighting environments composed of 10,000 or more objects. Seamless simulation requires user-friendly, self-configuring, variable-scale environments with essential resolution, and transparent connectivity. The IGSS program's intent to research areas of potential difficulty resulted in the selection of the following subprojects: (1) Integration of highly dynamic live objects with synthetic objects; (2) interoperability of coarse grain (e.g., time step wargames/aggregate units) with fine grain (realtime/individual units) simulations; (3) interoperability of engineering fidelity simulators with moderate fidelity simulators; and (4) networking of large numbers of objects (10,000 to 100,000) into one simulated warfighting environment.

An Advanced Interface Unit (AIU) will be developed to provide capabilities/tools for simulators and real systems to use in interfacing with the warfighting network.

This effort, called Highly Dynamic Vehicles (HYDY), Phase I, resulted in a Proof-of-Concept (POC) demonstration showing the feasibility of integrating a live F-14D aircraft into the simulation environment. Network connectivity was established between (1) an existing aircraft simulation facility located at the Naval Air Warfare Center Aircraft Division (NAWCAD) Manned Flight Simulator (MFS) in Patuxent River, MD; (2) the Naval Air Warfare Center Weapons Division (NAWCWD) System Integration Test Station (SITS) in Pt. Mugu, CA; and (3) the Secure Integration Simulation Laboratory (SISL) located at the Naval Command, Control and Ocean Surveillance Center (NCCOSC) Research, Development, Test and Evaluation (RDT&E) Division in San Diego, CA.

OVERVIEW

This paper provides a general discussion of the effort to connect the three facilities (MFS, SITS SISL), the modification made to those facilities to support this project, the connectivity established between and within the facilities, problems encountered, lessons learned, and the results of the HYDY Phase I POC demonstration.

The culmination of this effort will allow a live aircraft, while in flight, to interact Beyond Visual Range (BVR) with simulated aircraft (e.g., man simulators). This interaction will result from stimulation of the aircraft's radar and Radar Warning Receiver (RWR) based on an interchange of information between the simulated unit(s) and the live aircraft. The interchange will use ground communication of Protocol Data Units (PDUs) formatted in accordance with the Simulation Training and Instrumentation Command (STRICOM)/ARPA draft military standard for a Distributed Interactive Simulation (DIS) protocol and radio frequency (RF) communication using the "express PDUs." Based on information received in the PDUs, the radar and RWR will be stimulated. For this demonstration DIS PDUs were transmitted via ground-based communications facilities at the three sites specified above.

This effort demonstrated the ability to utilize DIS standard PDUs with HYDY vehicles (DIS entities) over a LAN and WAN by using one (or two) of these DIS entities to stimulate an active radar (APG-71) that was simultaneously receiving real aircraft returns. The challenge was to stimulate the radar such that the real and simulated returns were indistinguishable from each other.

FACILITY AND HARDWARE DESCRIPTIONS

Systems Integration Test Station (SITS)

The SITS facility develops, tests, integrates, verifies, and validates flight software for the F-14D fighter aircraft and provides a ground-based site for development and test of modifications required to allow injection of simulated air targets into the radar for display on cockpit display consoles. SITS contains the front section of an F-14D with all of the operational flight computers, the complete radar system, and a full cockpit. This partial airframe's

location allows the radar to be turned on and detect real aircraft flying on the NAWCWD test range. A Radar Target Stimulator (RTS) allows the injection of computer-generated targets into the radar system while detecting live targets on the range. Therefore, a live F-14D radar could be stimulated from a simulation network by modification of the RTS system, which would cause it to accept DIS PDUs as definitions of the targets it should inject into the F-14D radar system. The software is described in a later section.

SITS hardware connectivity starts with a local Navy Network (NAVNET) node (T1 tail circuit). NAVNET is a long haul T1 (1.544-million-bits-per-second [MBPS]) network with nodes located at various sites. The local NAVNET node feeds a 56-thousand-bits-per-second (KBPS) tail circuit into the SITS facility. The hardware connection inside the facility, from modem to F-14D airframe, is described in the following paragraphs.

Network Hardware Connectivity. Network connectivity consists of the following equipment required to bring the DIS PDUs to/from the AIU:

(1) A Codex 2172 Modem configured to operate at 56 KBPS connects the NAVNET tail circuit to a KG–84A.

(2) A KG–84A point–to–point data cryptographic device connects to the modem by a V.35 to RS–422 cable and a RS–422 to MIL–STD–188C Minimate connector. The V.35 connects to the modem while the MIL–STD–188C connects to the "Black" (encrypted) connection on the KG–84A.

(3) A TIMEPLEX Minilink unit provides the link between "phone" and computer communications. This unit is connected to the KG–84A "Red" (decrypted) connection by an MIL–STD–188C to RS–422 Minimate connector and a RS–422 to RS–422 cable.

(4) A RAD REB–10 ethernet bridge provides for a dedicated thick ethernet line to the AIU. This unit is connected to the Minilink via an RS–422 cable.

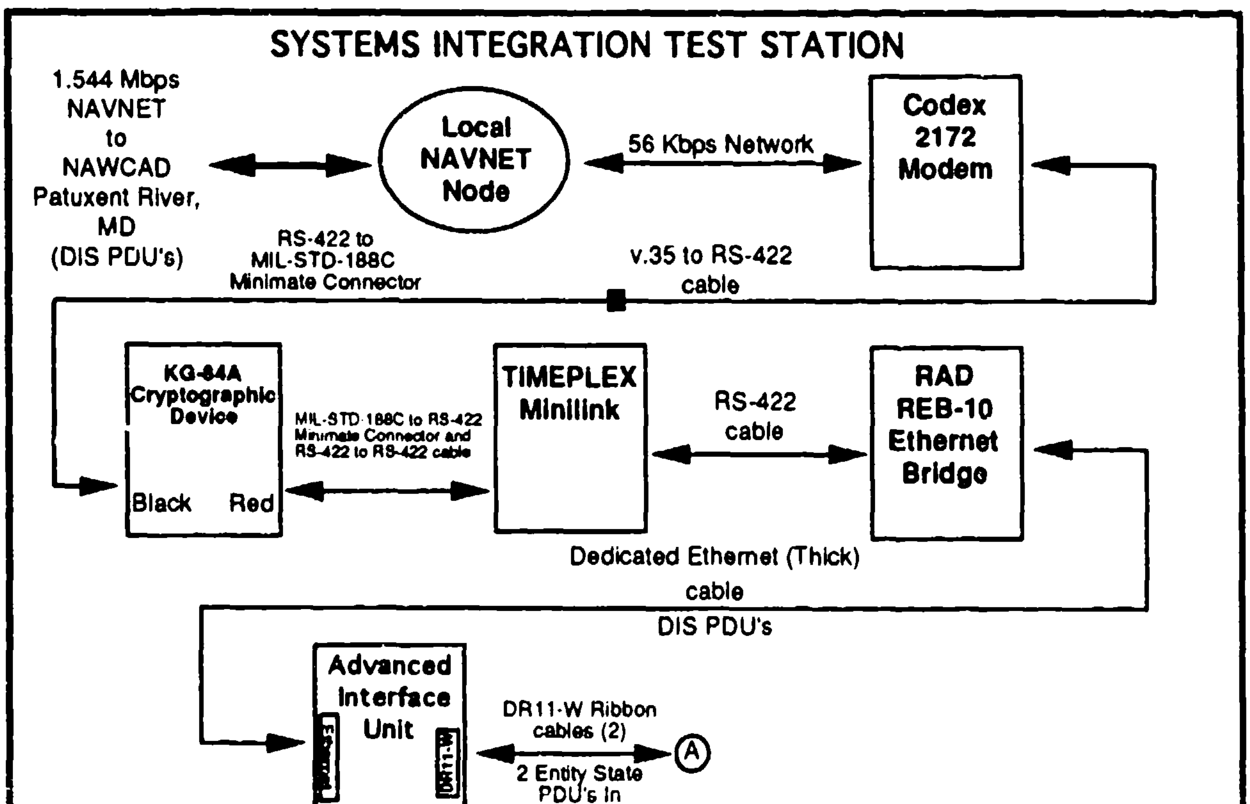


Figure 1. NAWCWD Hardware Connectivity

AIU Hardware Connectivity. The AIU is responsible for the network interface, the managing of DIS PDUs and the interface to the host, a VAX 8600 described in the next section. The AIU consists of a Force CPU-40 processor board and an Icron DR11-W emulator board, both residing in a VME card cage. The AIU is connected to the ethernet bridge via a thick ethernet cable. The DR11-W is connected to the VAX 8600 by two 16-pin flat ribbon cables; one for input and one for output of data.

The AIU also provides a simple PDU generation feature that allows the operator to cause the device to output PDUs to itself and to the network that represents an aircraft. This aircraft can be maneuvered interactively by the operator.

Radar Target Simulator Hardware Connectivity. The RTS hardware connectivity includes a VAX-8600 Interfacing Unit, the RTS system, and Airframe Interfacing Unit. These three "units" are described as follows:

(1) The VAX 8600 provides the interface between the AIU, RTS and the F-14D airframe. Thus,the AIU receives local entity data from the F-14D airframe interface and provides the RTS with the target entity data received from the network via the AIU. The VAX converts state information from DIS world coordinates to latitude and longitude and vice versa, compares the target/frame aspect ratios to generate target Radar Cross Sections (RCS), and formats RCS packet data to be sent to the RTS. It communicates with the AIU and the airframe interface via a DR11-W interface and with the RTS by "thin" ethernet. Each DR11-W interface requires two 16-pin flat ribbon cables.

(2) The RTS primarily consists of a Hewlett Packard HP-1000, a target generator chassis, RF and intermediate frequency (IF) interfaces, electronic countermeasures (ECM) equipment, and an IBM 386 personal computer for front-end operator interaction and display. The RTS provides for radar/target simulations and various electronics for IF link injection into the airframe radar. The front end for simulation control has a graphics display of what the radar sees. The RTS is connected via ethernet to the VAX 8600 and to the F-14D airframe via MIL-STD-1553B cables (to the radar bus) and IF link.

(3) The airframe interface consisting of a Force CPU-40 processor, an Icron DR11-W emulator, and a 1553B bus interface board all residing in a VME card cage. This unit provides airframe state vector and orientation data to the VAX-8600. The DR11-W interface connects to the VAX-8600 via the two 16-pin flat ribbon cables and the 1553 bus interface connects to the airframe.

F-14D Airframe/Test Management Station H/W Connectivity

(1) The Test Management Station (TMS) is the "software pilot" that "flies" the F-14D airframe. The TMS provides the airframe equipment with all the dynamic flight data necessary to indicate in flight conditions. The TMS is connected to the airframe via MIL-STD-1553B cables (to the mission computer and radar buses) and various discrete connections.

(2) The F-14D airframe platform receives "flight" data and characteristics from the TMS and radar target data from the RTS. This is an actual F-14D cockpit with all required equipment for radar operations and analysis.

Equipment Present for Demonstration. For the purposes of the demonstration, the following nonstandard equipment (in addition to the AIU) was connected to the ethernet network that provided the interface between the REB-10 and the AIU.

(1) The Technologies System Inc. (TSI) Low Cost Stealth was connected to provide for both two- and three-dimensional displays of the simulated world as represented by the PDUs received over the network. This includes PDUs generated to report the position of the SITS airframe (pseudo live aircraft) and any simulation entities being reported on the network.

(2) The SIMULIZER was connected to provide a scripted "target" generation capability. The SIMULIZER would output a scripted set of DIS PDUs on command that represented one or more aircraft flying a predetermined route.

Manned Flight Simulator

The MFS facility's mission is to provide, through simulation, test and evaluation (T&E) of aircraft and onboard aircraft avionics systems and pilot training. The MFS includes high-fidelity flight-dynamics system simulation, avionics system simulators, a wide field-of-view (FOV) visual system for man-in-the-loop evaluations, and a motion-base system to provide acceleration cues for conventional takeoff and landing tasks as well as vertical and short takeoff and landing, hover, and transition. The MFS incorporates four simulation stations: the motion base, the dome, the laboratory station, and the crew station which includes the F-14 back seat, the F/A-18, and the V-22 Government Test Pilot Trainer. A COMPU-SCENE IV Visual Image Generator (VIG)

system and an IRIS visual system provide high- and medium-resolution visuals. In addition, the dome FOV simulator provides two projectors for target generation.

The dome with an F/A-18 and one target generator was used for this POC demonstration. The simulation program at MFS was modified to accept DIS PDUs for target generation and to output DIS PDUs from the simulation platform. The software is described in a later section.

NAWCAD hardware connectivity starts with a local NAVNET node (T1 tail circuit). The local NAVNET node feeds a 56-KBPS tail circuit into the MFS facility. From inside the facility, the hardware connection from modem to cockpit simulator is described in the following paragraphs.

Network H/W Connectivity. Network connectivity consists of all the following equipment required to bring the DIS PDUs to/from the AIU.

(1) A TYTEK Modem configured to operate at 56 KBPS connects the NAVNET tail circuit to KG–84As.

(2) Two KG–84A point-to-point data cryptographic devices connect to the two modems by RS–449 to RS–449 cable and RS–422 to MIL–STD–188C Minimate connector. The RS–499 connects to the modem while the MIL–STD–188C connects to the "black" (encrypted) connection on the KG–84A. Two KG–84As are needed as one is connected to SITS and the other is connected to SISL.

(3) A TIMEPLEX Minilink unit provides the link between "phone" and computer communications. This unit is connected to the KG–84A on the "red" (decrypted) connection by a MIL–STD–188C to RS–422 Minimate connector and from the Minimate to the Minilink by a RS–449 to Minilink tempest cable.

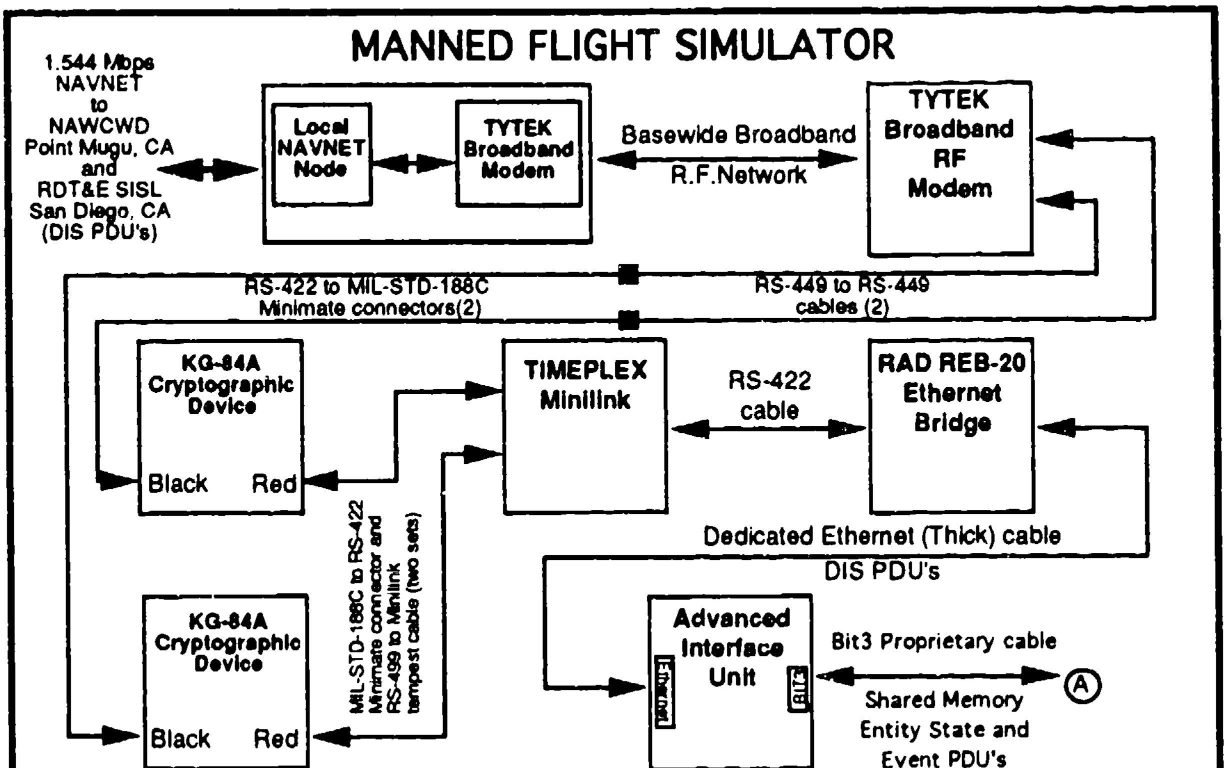


Figure 2. NAWCAD Hardware Connectivity

(4) A RAD REB–20 Ethernet Bridge will pass ethernet packets to and from each of the NAVNET connections and provide for a dedicated thick ethernet line to the AIU. This unit is connected to the Minilink via an RS–422 cable.

AIU H/W Connectivity. The AIU is responsible for the network interface, the managing of DIS PDUs, and the interface to the Host, a Digital Equipment Corporation (DEC) Micro VAX II. The AIU consists of a Force CPU–40 processor board and a Bit3 Qbus-to-VME adapter board with 8-Mbytes of dual-port RAM, both residing in a VME card cage. The Bit3 card allows the CPU–40 and a Micro VAX to share one Mbyte of RAM. The AIU is connected to the RAD REB–20 Ethernet Bridge via a thick ethernet cable. The Bit3 card is connected to the Micro VAX II by the Bit3 supplied flat ribbon cables; one for input and one for output of data.

Cockpit Simulator H/W Connectivity. The cockpit simulator consists of

(1) Micro VAX (uVax) providing the AIU "host" processes. This process manages the transfer of DIS PDUs between the AIU and multiport memory. The uVAX contains the Qbus side of the Bit3 Qbus-to-VME Adapter to the AIU.

(2) uVax dedicated for avionic simulations. The F/A–18 avionics models duplicate the functions performed by the actual avionics in the aircraft. The avionics models simulate the AYK–14 mission computers and display processors. The uVax simulations receive/send information from/to the multiport memory and 1553 bus.

(3) VAX 8650 used for aerodynamic simulation (airframe), visuals and coordinate conversions. This VAX maintains a high-fidelity aerodynamic model of the F/A–18. The visual process reads and processes incoming "flat earth" data from the entity table in Multiport Memory. The coordinate conversion process performs transformations to/from DIS world coordinates and MFS simulator flat-earth coordinates. This VAX communicates to the Avionic Simulations uVAX (through multiport memory) and to the GE COMPU–SCENE IV.

(4) Multiport memory providing a common data link (shared memory) to avionics and flight dynamics models.

(5) GE COMPU–SCENE IV VIG system to provide for the visual environment in the 40-foot dome surrounding the cockpit and target projector.

(6) uVAX used for communication with the cockpit via MIL–STD–1553 interfaces.

(7) fully functional cockpit of an F/A–18 Hornet.

Secure Integration Simulation Laboratory

The SISL facility is a secure environment for the development, test, and integration of the AIU software. Access is provided to development workstations, AIU hardware, and communications channels for interacting with remote simulations or other AIU devices. The cryptographic equipment allows interaction with the MFS facilities. Interactions between SISL and SITS were constrained to go "through" the MFS connection as described above.

SISL hardware connectivity starts with a local NAVNET node (T1 tail circuit), which feeds a 56-KBPS tail circuit into the SISL facility. The hardware connection inside the facility, from modem to the laboratory network, is described in the following paragraphs.

Network Hardware Connectivity. Network connectivity consists of all the following equipment required to bring the DIS PDUs to/from the laboratory network:

(1) A NEC Modem configured to operate at 56 KBPS connects the NAVNET circuit to a KG–84A.

(2) A KG–84A point–to–point data cryptographic device connects to the modem by a V.35 to RS–422 cable and a RS–422 to MIL–STD–188C cable. The V.35 connects to the modem while the MIL–STD–188C connects to the "black" (encrypted) connection on the KG–84A.

(3) A TIMEPLEX Minilink unit provides the link between "phone" and computer communications. This unit is connected to the KG–84A "red" (decrypted) connection by an MIL–STD–188C to RS–422 cable.

(4) A RAD REB–1 Ethernet Bridge provides for a dedicated thick ethernet line to the AIU. This unit is connected to the Minilink via an RS–422 cable.

Laboratory Network Equipment. The SISL facility contains:

(1) Bolt, Beranek, and Newman (BBN) Semi-Automated Force (SAFOR) system for generation and control of simulation entities and transmission of the entities on the network.

(2) TSI Low Cost Stealth for display of two- and three-dimensional views of the simulated world.

(3) TSI translator for Simulation Network (SIMNET)/DIS protocols.

(4) AIU GenTrack unit for generating test entities in either SIMNET or DIS protocols.

(5) Data logger for recording of DIS or SIMNET PDUs to allow playback and/or analysis or testing, demonstrations, and exercises.

NETWORK CONNECTIVITY AND SECURITY

Long-haul network connectivity and security were major issues during the preparation for this POC demonstration. Secured network connectivity between the SITS, MFS and SISL was not established by the time of the POC. The connectivity problems are believed to be the result of KG-84A "strapping" problems. (Each of the four KG-84As needs to be configured or "strapped" before use.)

Originally the POC was to use the secure Distributed Simulation Internet (DSI) network to provide a multipoint connection network with no single point of failure. However, the secure network capabilities were not installed at all three sites by the time of the demonstration.

As an alternative, the U.S. Navy's NAVNET was selected and KG-84As were used as the encryption devices. Because the KG-84 is a point-to-point encryption device, one site must serve as the "hub" of the network and "forward" PDUs for the other two sites. The MFS served as the hub for the POC network. All PDUs generated at the SITS and the SISL were to be sent to MFS; MFS was to send its PDUs to both SITS and SISL. In addition, MFS was to forward all SISL generated PDUs to SITS and forward all SITS generated PDUs to SISL. However, this setup was not successfully installed in time for the POC demonstration.

The problems encountered in getting the network in place and tested were the required long lead time for installing the communications circuits; developing and getting approval of the security procedures and documentation at three sites; generating Memorandum Of Agreements (MOAs), staffing them through the local approval chains and getting the Designating Approval Authority (DAA) to sign off on the network; and getting the required strapping and correct cabling at all three sites as each site has a different modem.

As a result, technicians were still trying to "strap" the 4 KG-84As, at geographically dispersed sites on the day of the POC demonstration. Because of the difficulties of coordinating this strapping effort between three different commands in two different time zones, the secure network was not successfully established. Point-to-point connectivity was successfully established among pairs of sites but not between all three sites simultaneously.

SOFTWARE DEVELOPMENT AND MODIFICATION

This effort required the generation of new software and the modification of existing software. Software for the AIU devices was developed at NCCOSC as enhancements and modifications of software developed during previous efforts. The AIU hardware consisted of two slightly different configurations for the MFS and SITS sites. The software developed was also slightly different. In addition to the AIU software, both MFS and SITS had to develop new software and modify existing software to allow their systems to communicate with the AIU devices.

SITS Software

The existing software was modified to interface with the AIU. These modifications allowed the generation of entities within the RTS based upon Entity State PDUs (ESPDUs) received from the AIU. In addition, the software was modified to output an ESPDU (representing the F-14D) to the AIU. The AIU software was modified to ignore events from the network (Fire PDUs, etc.). The AIU transfers up to two ESPDUs to the SITS system for use in stimulating the RTS.

SITS AIU Software. The AIU reads and writes DIS PDUs to the ethernet port on the Force CPU-40 card; maintains table of entities and their current DIS information, filters entities, transmits selected entities to SITS; ignores all DIS event PDUs; filters out and ignores all non-DIS ethernet packets; maintains SITS F-14D in DIS based on data received from SITS; and tests entity generation.

The interface between the AIU and SITS is maintained by the exchange of data at the rate of four times a second (4 Hertz) via Direct Memory Access (DMA) and DMA emulation. This is implemented by the DR11-W cards. The AIU and SITS read and write data to/from each other's memory via the DR11-W cards. The SITS performs actual DMA transfers while the AIU performs DMA emulation.

The AIU, when interfacing with the DR11-W card, performs byte swapping and floating point conversions. The byte swapping is necessary when communicating data to/from the VAX due to different processor architectures. The floating-point conversions were required to convert IEEE floating-point representation in the DIS PDUs to/from VAX floating point.

All DIS PDUs that are not ESPDUs are ignored. The incoming ESPDUs are first passed through a filter that drops any ESPDUs that are not located within a 70-mile cylinder projected 60 miles in front of the

SITS F–14D aircraft entity. ESPDUs that fall within the filter are then copied into a hash table with the location determined by the ESPDU entity identification field. Since the RTS can accommodate only two targets per run, the AIU sends only the first two ESPDUs it receives to the RTS.

The SITS system transfers an ESPDU for the F–14D to the AIU via the DR11–W interface. The AIU transmits the first ESPDU received from SITS and then dead reckons the SITS entity and outputs a new ESPDU whenever the SITS entity's position or orientation differs from the dead-reckon values by a preset (and modifiable) location and/or orientation threshold. If no positional threshold has been exceeded (i.e., no ESPDU sent out) for 5 seconds, then one is output as required by current convention.

Also, software was built into the AIU to allow generation of a test entity. Through an operator interface, the AIU could be instructed to generate an ESPDU for test purposes. This software would allow the generation of an ESPDU representing an F–14D aircraft entity type and allow modification of the entities location, altitude, course, and speed.

SITS System Software. The SITS system software obtains F–14D airframe state vector data; provides transformations of coordinate, velocity and orientation of the incoming and outgoing ESPDUs; calculates RCS for all target entities; and provides interface processing to the airframe, RTS, and AIU.

The SITS system software consists of the airframe interface, the RTS interface, and the VAX process. The airframe interface software was written to extract the airframe state vector data from the mission computer bus via a MIL–STD–1553 interface and send the data to the VAX via a DR11–W interface. The RTS interface was added to the existing RTS and was provided to receive target RCS data from the VAX via Ethernet connection and to use this data instead of the usual "canned" target data that the RTS was initially designed to use. The VAX process ties the AIU, airframe and RTS interfaces together.

The VAX, utilizing modified software from an existing aerodynamic model, provides for all coordinate, velocity and orientation transformations and calculates the target RCS data. The VAX also controls all interface timing and communication between the VAX and the AIU, airframe interface and RTS. The VAX sends and receives ESPDUs to/from the AIU via a DR11–W and receives airframe state vector data from the airframe interface via another DR11–W interfaces and sends the RCS data to the RTS via ethernet "thin" connection.

MFS Software

At MFS, the existing software was modified to interface with the AIU, which allowed the generation of entities within the simulator based upon ESPDUs received from the AIU. The AIU software was modified to send the event PDUs; Fire, Detonation, and Collision. The MFS simulator software will receive the events but not currently process them. In addition, the software was modified to output an ESPDU (representing the simulator) to the AIU. The AIU maintains a table of all entities for the MFS to interrogate and two ring buffers; one containing new entity messages and one containing events, and outputs the simulator's ESPDU on to the network.

MFS AIU Software. The AIU reads and writes DIS PDUs to the ethernet port on the Force CPU–40 card; maintains a table of entities and their current DIS information; passes new entity notifications to MFS; passes DIS Fire, Detonation, and Collision event PDUs to MFS; filters out and ignores all non-DIS ethernet packets; maintains MFS simulator in DIS based on data received from MFS; and generates test entity.

The interface between the AIU and MFS is maintained by data located in shared memory and is implemented by the BIT–3 cards. The AIU and MFS read and write data to a dual port memory maintained on the VME side of the BIT–3 cards. The MFS interfaced with the memory on its BIT–3 card in the same manner as it interfaces with normal VAX memory. The AIU interfaced with its BIT–3 memory card in the same manner as any offboard (nonlocal) VME memory with the exception of data representation.

The AIU, when interfacing with the BIT–3 card, performs byte swapping (necessary when communicating data to/from the VAX due to different processor architectures), and floating-point conversions (required to convert IEEE floating-point representation in the DIS PDUs to/from VAX floating point format)

In the BIT–3 memory, the AIU maintains a table of 1033 "slots" for input DIS entities. Entities are inserted into the table based on a hashing algorithm. When a new entity is received from the network, its entire ESPDU is inserted in the table and the MFS notified by a message in a ring buffer (also in BIT–3 memory). When a new ESPDU is received for an existing entity, the data overlay the existing ESPDU. The MFS can interrogate individual entries in the table whenever it requires data. The AIU dead reckons table entries by updating location and time

stamp information in the table approximately 10 times a second.

Also in the BIT-3 memory is a table maintained by the MFS with 19 slots for MFS entities. The MFS updates these entities at its own processing rate. Three ring buffers in BIT-3 memory transfer event data; one for each direction and one for notification of new ESPDUs to MFS. The ring buffers are maintained as first-in-first-out (FIFO) queue with the oldest message being overlaid in the event of buffer overflow. All DIS PDUs that are not ESPDUs are inserted into the to MFS ring buffer when received. A second ring buffer transfers event PDUs from the simulator to the AIU. The third ring buffer passes indications of new ESPDUs received by the AIU to the MFS.

The AIU transmits an ESPDU at the first encounter in the MFS entity table. The AIU dead reckons the MFS entity based upon the ESPDU and outputs a new ESPDU whenever the MFS entity's position or orientation differs from the dead reckoned values by a preset (and modifiable) location/orientation threshold. If no threshold has been exceeded (i.e., no ESPDU output) for 5 seconds, then an ESPDU is output. Software built into the AIU allows generation of test entities. An operator interface would allow the software to generate ESPDUs representing a user specified entity type at a user specified location, altitude, course, and speed.

MFS Simulator Software. The MFS simulator software obtains incoming event and entity state PDUs from the AIU places them in multiport memory, and provides transformations of coordinate, velocity, and orientation of the incoming and outgoing ESPDUs.

The host process software provides the MFS interface to the AIU, retrieves the incoming DIS PDU data and puts it into the multiport memory, extracts local entity data and events, and places the data into the AIU shared memory. The visual process reads the entity table in the multiport memory, applies coordinate transformations to flat earth, and provides the GE COMPU-SCENE IV with the entity data for target generation. The local entity process performs coordinate transformations on the F/A-18-state vector data and provides this data to the multiport memory, which is read by the host processes.

TEST AND INTEGRATION EFFORTS

Individual pieces were tested separately to the extent possible. Next, the AIU was installed at the MFS site and integration testing with the MFS system was performed. By using the lessons learned at MFS, the AIU was installed and integrated into the system at SITS. However, a communications network had to be established and approved for secure operations before full system integration could be accomplished. The available time expired before the network was successfully used as a transparent encrypted communication medium.

SITS Testing

SITS testing consisted of installing the AIU, establishing DR11-W communications, defining proper data representation (byte swapping and floating point formats), and the sending and receiving of ESPDUs. Interaction of the AIU operating system/CPU board interaction with the Icron DR11-W emulator board resulted in timing problems that took up the bulk of the AIU/SITS interface testing. Proper data representation was accomplished utilizing built-in test DIS PDUs.

To facilitate testing without input from the network, the AIU was modified to generate a F-14D ESPDU, which was injected at the network handler level so as to appear to the AIU/SITS as a real entity. The test ESPDU was successfully displayed on the F-14D APG-71 radar.

A second means of generating an ESPDU was brought online by the Unix Workstation based SIMULIZER. The SIMULIZER generated ESPDUs following a predetermined script, which proved useful for testing consistent fixed flight patterns.

SITS testing also included F-14D "rollout" exercises where the SITS doors were opened and the cockpit was rolled out to allow the radar to actively radiate. Due to close proximity to major airports, many "targets of opportunity" were available for tracking, which allowed for a variety of "real" vs. "simulated" radar operations.

MFS Testing

MFS testing consisted of installing the AIU, configuring the BIT-3 cards, defining proper data representation (byte swapping and floating point formats) and the sending and receiving of ESPDUs. Configuring the BIT-3 required some calls to the BIT-3 corporation. Proper data representation was accomplished utilizing built-in test DIS PDUs.

To facilitate testing without input from the network, the AIU was modified to generate an operator specified entity and allow operator dynamic interaction. The ESPDUs injected at the network handler level, appeared to the AIU/MFS as real entities, which proved to be invaluable in MFS testing as no network traffic was available. Utilizing these test ESPDUs, target visuals were successfully displayed in the simulator dome.

MFS testing also included "wrapping around" the F/A–18 MFS entity with an offset back to the target generator. Thus, the AIU/MFS loop was tested with the target visual acting as a wingman following the F/A–18 movements.

AIU Testing

Both AIU systems, with the exception of correct byte swapping, were extensively tested. A "host" process was designed and coded to run on a separate processor board to simulate/stimulate the SITS DR11–W interface and VAX host process. For the MFS, a host process board and a shared memory board were used to test the AIU against the shared memory, ring buffers, and VAX host processes. Another separate processor board AIU with DIS PDU generation capabilities along with built-in ESPDU test generators were used to generate incoming and outgoing PDUs. "Canned" DIS PDUs were built into each AIU expressly for byte swapping testing.

SISL Testing

Testing at the SISL laboratory occurred in stages:

(1) Final connectivity checks between all systems occurred.
   - BBN SAFOR to TSI (SIMNET 6.6.1 protocol)
   - GenTrack to TSI (DIS 1.0 protocol)
   - GenTrack to Data Logger (DIS 1.0 protocol)
   - Data Logger to TSI (DIS 1.0 protocol)

(2) Once all connections were verified, a F–14D was created on the SAFOR workstation and receipt of the PDUs checked at all other devices. This test failed because the TSI translator was not functioning properly.

(3) DIS F–14D aircraft was verified to be sending out to the network but time limitations and security problems precluded a full scale test.

End-to-End System Integration

The goal was to establish a secure network between SITS, MFS, and SISL and then show interaction between all entities operating concurrently. The first step was to establish the secure communications network that would be transparent to the systems using it. Overall systems integrations testing could be performed in accordance with the "Naval Air Warfare Center Aircraft Division (NAWCAD) and the Naval Air Warfare Center Weapons Division (NAWCWD) IGSS Lab Assessment and Procedures." The unsecured network communications were established and verified using the AIU PDU generation facilities at MFS and SITS, the BBN SAFOR PDU generation facilities at SISL, the TSI Low Cost Stealth for display at SISL and SITS, and the MFS dome simulator for display at MFS. Attempts were made to establish secure communications between all three sites using the same equipment to transmit and verify receipt of PDUs. Two-way secure communications were established between SISL and MFS and between MFS and SITS. Three-way communications were not successful, which meant integration testing was not possible.

PROOF-OF-CONCEPT DEMONSTRATION

The HYDY Phase I POC (November 1992 at NAWCWD) consisted of an In-Progress-Review covering both what had been accomplished for phase I, plans for phase II, and demonstrations in the SITS laboratory. Simultaneous radar stimulation with simulated and real targets was the main goal for HYDY Phase I and was successfully demonstrated. The SITS F–14D airframe was "rolled out" on both days of the POC demo to allow the APG–71 radar to actively radiate for detection of live aircraft. The two simulated targets were generated by the SITS AIU and the SIMULIZER.

The first day rollout was marred by a loose 1553 bus connection at the airframe. The bad connection caused noise on the line that wreaked havoc with the interface to the TMS (the program that "flies" the F–14D airframe), and the RTS. These programs would not run long due to lack of stable data from the frame. Eventually, a simulated target successfully injected into the radar receiver resulted in the track displayed on the Radar Intercept Operator (RIO) console.

The second day rollout was more successful. Simulated tracks (two) along with real aircraft were tracked on the RIO console. The simulated tracks were run through a variety of course and speed changes (altitude was not changed). The live aircraft flew "race track" patterns with a long leg coming in off the ocean directly at the SITS laboratory. This flight path, repeated at regular intervals, made it possible to generate a simulated entity that appeared to be flying with the live aircraft and showed the simultaneous detection and tracking by the F–14D radar of both live and simulated aircraft. The demonstration showed no detectable difference between the radar returns of a real aircraft and a simulated aircraft.

The lack of a secured line prevented the MFS dome simulator from providing a manned simulator to stimulate the SITS. The network demonstration

was to have the two facilities, SITS and MFS, "fly" against each other. An example scenario would have put the MFS's F/A-18 cockpit BVR in front of the SITS F-14D airframe with a closing course for a fly by. The F-14D would pick up the F/A-18 as a track on its radar and the MFS would visually display the F-14D when the aircraft came within visual range. As it turned out, both sites had to fly against locally generated AIU DIS aircraft entities. The AIU provides the DIS network compatibility required for each of the sites and has a built-in test function that allows them to generate a variety of DIS entities.

FUTURE EFFORTS

Once the secure communications network is fully functional, the testing and integration will be completed. Work will continue toward building a more robust AIU by using the lessons learned during this effort and research in other areas.

The effort to interface with the F-14D RADAR system will continue with the development of an Airborne RTS (ARTS). This effort will reduce the size of the current RTS and combine the functionality of the AIU into the ARTS hardware suite. The airborne version of the AIU will be modified to accept input from a radio receiver vice an ethernet input. The PDUs received by the radio will be communicated to the ARTS via a 1553B bus. The content of the ESPDUs will be modified because of the limited bandwidth of the transmission hardware. Current plans call for the PDU structure to reflect the design put forth in "Analysis of DIS Protocols for DARPA'S HYDY Project." The generation of the ESPDU representing the live aircraft will be done by a ground station using current Tactical Aircrew Combat Training System (TACTS) range ground station technology and in the air by the airborne AIU. The ground station will serve as an AIU for airborne entities outputing full DIS ESPDUs for the aircraft, filtering received PDU traffic, formatting PDUs for transmission to the aircraft, and conveying the PDUs to the radio for transmission.

In parallel with the above effort, the secure network will be established. The three sites will be integrated and used to perform data latency testing to determine the effects of transmission delays over long distances when using high-fidelity simulations. This may result in further changes or enhancements to the AIU to mitigate any deleterious affects of transmission delays.

REFERENCE

"Military Standard Version 1.0 (Draft) - Protocol Data Units for Entity Information and Entity Interaction in a Distributed Interactive Simulation," Institute for Simulation and Training, University of Central Florida, October 1992.

ACRONYMS

AIU–Advanced Interface Unit
ARPA–Advanced Research Projects Agency
ARTS–Airborne RTS
BBN–Bolt, Beranek, and Newman
BVR–Beyond Visual Range
DAA–Designating Approving Authority
DIS–Distributed Interactive Simulation
DMA–Direct Memory Access
DSI–Defense Simulation Internet
ECM–Electronic Counter Measures
ESPDU–Entity State PDU
FOV–Field-of-View
HYDY–Highly Dynamic Vehicles in a Real and Synthetic Environment
IF–Intermediate Frequency
IGSS–Intelligent Gateway/Scaleable Simulation
IPR–In–Progress–Review
KBPS–Thousand Bits Per Second
LAN–Local Area Network
MBPS–Million Bits Per Second
MFS–Manned Flight Simulator
MOA–Memorandum of Agreement
NAVNET–Navy Network
NAWCAD–Naval Air Warfare Center Aircraft Division
NAWCWD–Naval Air Warfare Center Weapons Division
NCCOSC–Naval Command, Control and Ocean Surveillance Center
PDU–Protocol Data Unit
POC–Proof-of-Concept
RCS–Radar Cross Section
RDT&E–Research, Development, Test and Evaluation
RF–Radio Frequency
RTS–Radar Target Stimulator
RWR–Radar Warning Receiver
SAFOR–Semi–Automated Forces
SIMNET–Simulation Network
SISL–Secure Integration Simulation Laboratory
SITS–System Integration Test Station
STRICOM–Simulation, Training, and Instrumentation Command
TACTS–Tactical Aircrew Combat Training System
TMS–Test Management Station
TSI–Technologies System Inc.
VAX–Virtual Address Extension
VIG–Visual Image Generator
VME–Versa Modula Europa
WAN–Wide Area Network

An Object-Oriented Approach to Environment on the Virtual Battlefield

Rosemary Enright
SYSCON Corporation
Middletown, Rhode Island

Randal Holl
SYSCON Corporation
New London, Connecticut

ABSTRACT

Wargames and warfare trainer systems require varying degrees of environmental fidelity depending on their focus. Individual combatant trainers need a high measure of congruity between the real and perceived environment; strategic planners achieve a realistic overall view of the battlefield with fewer details. In fact, excessive environmental fidelity may be detrimental to the training mission. We have been working to create a model of the environment that may be tailored to the level of abstraction appropriate for the simulation. We describe the environment as a series of objects with the attributes and services needed to calculate mobility and detection. For our current wargame, we have concentrated on the requirements of strategic planners, while identifying the characteristics and methods required to extend the concept to the battle force level and below.

ABOUT THE AUTHORS

Rosemary Enright joined SYSCON Corporation in 1981 as part of the team designing the Naval Tactical Game (NAVTAG), a time-stepped tactical trainer for surface ship Tactical Action Officers. She has since worked on the S14A13 team trainer and on acoustic performance prediction systems, including the AN/UYQ-25A (SIMAS). She is currently Lead Requirements Analyst on a multiservice, strategic level wargame. Ms Enright holds a Masters of Arts degree from New York University and prior to joining SYSCON taught at New York Institute of Technology.

Randal Holl is manager of SYSCON Corporation's Systems Engineering Program Unit in New London. He has been involved with Navy training systems for over eight years and managed the development of the Tactical Advanced Simulated Warfare Integrated Trainer (TASWIT). Previously, Mr. Holl served as a Lieutenant in the U.S. Submarine Fleet. He holds a Masters of Science degree in Computer Science from Rensselaer Polytechnic Institute and a Bachelor of Science degree in Physics from the Massachusetts Institute of Technology.

An Object-Oriented Approach to Environment on the Virtual Battlefield

Rosemary Enright
SYSCON Corporation
Middletown, Rhode Island

Randal Holl
SYSCON Corporation
New London, Connecticut

INTRODUCTION

Increasingly complex missions, reduced Optempo, and increased coordination between friendly forces have greatly expanded the demands made upon warfighting simulations. The new generation of simulation systems must address the roles of training, mission rehearsal, and operations planning. In the long term, research and development, acquisition, and combat system effectiveness may also depend on these simulations. Meeting this wide range of demands will require the simulation of large numbers of warfighting entities from the independent unit level through several levels of aggregation to full scale warfare planning. As figure 1 shows, current systems address only pieces of the problem.

Computer and communications resources now have the capacity to host the required complex simulations. These resources are, however, neither abundant nor easily integrated and, as the past can attest, requirements rise steadily to test the limits of available resources. "Brute force" cannot and should not be used to overcome all obstacles. Finesse in the design and execution of simulation protocols, the use of communications bandwidth, and the notion of "fidelity" as applied to the requirements and goals at each level of aggregation is a necessity.

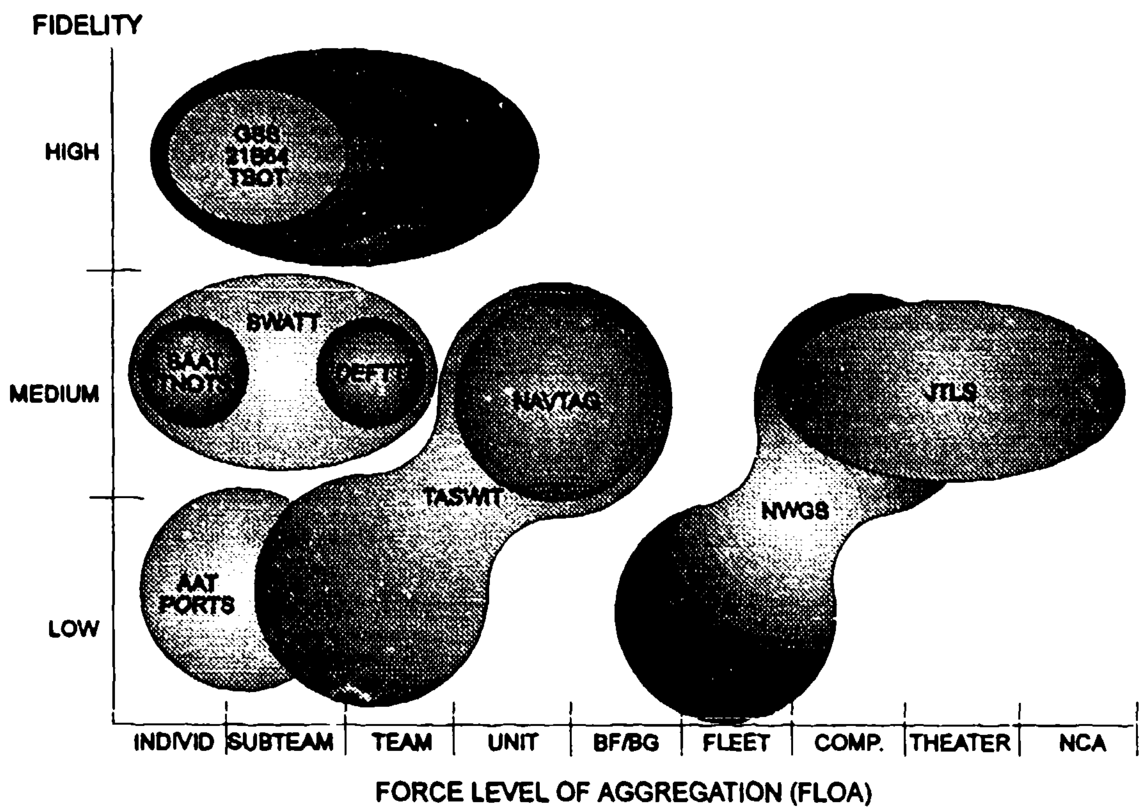


Figure 1. Current Military Training and Gaming Systems

MODELING CONSISTENCY

The availability of computer and communication resources has given rise to concern about modeling consistency and ultimately about the applicability of models to specific uses.

Distributed Simulation

In a distributed simulation, separate processors are independently computing and displaying the effects of interaction between a set of entities. In such simulations, each processor must produce output that avoids discontinuities in "ground truth" or differing world views; that is, all of the processors must "play" in the same virtual world.

Distributed Interactive Simulation (DIS) networks have the added complexity of linking disparate simulators each of which has been independently designed and built to meet a unique set of requirements. For effective integration of such simulators in a distributed simulation, not only must model outputs be consistent, but the outputs must also produce consistent effects. For example, even with identical output, tank operators working with a relatively low granularity display consistently outperformed tank operators using a simulator that employed high performance displays. The camouflage and background visuals rendered on the high performance display were more realistic and, therefore, screened the opposing force more effectively.

Physical vs Statistical Models

Additional "world view" variances can arise between simulators operating at different levels of aggregation. Single entity simulators, like the tank simulators mentioned above and flight training simulators, model the world on a physical basis. Effects that impact the operation or appearance of the entity are modeled using equations that mimic the physics governing the behavior of the entity or the environment. Force, mass, acceleration, and line of sight are typical of the parameters involved.

At a high level of aggregation, however, the modeling is quite different. Here the approach usually taken is to model behavior, typically interaction outcome, as a sample drawn from a statistical distribution characterizing all possible outcomes. The parameters involve sets of probability density functions and their respective parameters of variation.

COMPLEXITY VS GRANULARITY

How can this obvious disparity be rectified? Can one modeling approach be extended to cover the range of the other? Perhaps, but only at great cost in compute power and model complexity.

Figure 2 shows the problem with using a single modeling approach across the full spectrum of aggregation levels. The use of physical models prevents computation on an aggregated force basis. Each element of the force must be modeled in detail for a physical model to make sense. If a rain squall blankets half of a battlefield, for example, the physical models for detection must know where the element detected is relative to the intensity of the rain in order to calculate the individual detect/non-detect outcome. Combinatorial explosion occurs as the number of entities increases.

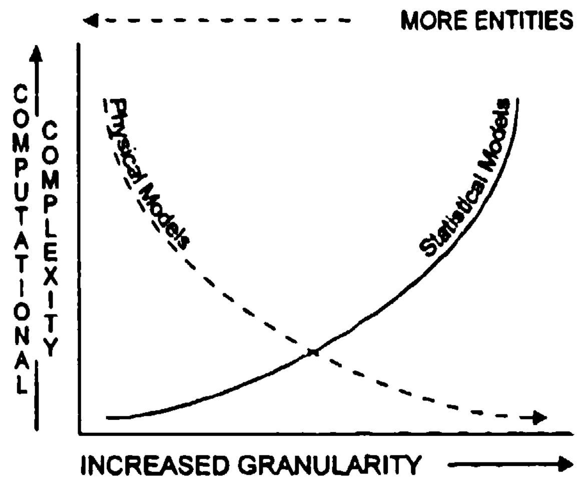


Figure 2. Computational Complexity

Likewise for statistical models. As the force deaggregates and focuses less on an outcome and more on a detailed view of the interaction, the number of statistical parameters required to support the granularity undergoes a similar combinatorial explosion.

Even if physical models can be employed to compute the interaction of an aggregated force on an element by element basis, such an implementation can be detrimental to the

objectives of aggregate simulation. As the level of aggregation increases, the purpose of a simulation shifts away from the procedural and tactical towards the strategic. At the strategic level, fused information rather than individual data points is required. Physical modeling greatly reduces the capacity to "speed up" the game to show this fused information at greater than real-time rates and introduces the need for fusing models. In addition, the single data point calculated by the physical model may be less representative of the overall probability as provided by the statistical model's density function. For training, this possible skewing, as well as the introduction of detail that is not the direct subject of the training, tends to distort the intended lesson. For operations research analysis, incorrect inferences may be drawn.

ENVIRONMENTAL MODELING

The consistency and complexity problems introduced by advances in distributed simulation have led to the examination and isolation of the elements that create a seamless battlefield - among them the environment. Environmental modeling is one of the most demanding challenges facing simulation design. Armed conflict spans all terrain from mountain to jungle to desert, and respects no boundary of air, sea, land, or, increasingly, outer space. Environmental factors impact and often times dictate how the battle is waged. Even the most fundamental and global parameters of the environment are highly complex; complexity is then increased by local effects such as fog, wind, and magnetic variations.

The good news is that algorithms requiring environmental data tend to follow relatively simple processing paths. Detection processing, a central algorithm for any warfare simulation, is typical. Detection starts with one or more observable parameters from the object to be observed, e.g., signal level. The value of this observable parameter is modified by the environmental medium between the observed and the observer, but the parameter units do not change. For example, the radiated acoustic noise from a submarine is dispersed, reflected, refracted, and attenuated in its travel through the ocean to a receiving sonar. A 100 dB emission may be only 50 dB at the receiver, but it is still an acoustic noise level. In a low aggregation simulator, the receiving sonar system simulation converts this signal to a human-oriented input, usually some sort of visual display. At higher levels of aggregation, a detect/no-detect decision may be calculated from a statistical distribution function. At the highest level of aggregation, the output may be the probabilistic outcome of the interaction in terms of kill or be killed.

All aggregation levels follow the same processing pathway from emission, through the environment, to an output. As a result, one can potentially design a set of "plug replaceable" environmental models within a single simulation framework that are selectable based on level of aggregation.

OBJECT-ORIENTED ENVIRONMENT

We have been working towards the design of such an approach using the natural advantages of object-oriented (OO) methodologies. We believe that this approach can provide a seamless progression through the aggregation spectrum and allow effective interaction between simulators at adjacent aggregation levels.

The use of OO allows us to embed several levels of environmental models into a single object structure through the use of inheritance and operation overloading. Each subclass inherits all the characteristics (attributes) and operations (services) of its parent class. The subclass can add attributes and services and can also overload the services, changing the processing that is performed. We have defined classes that contain the minimum attributes and services needed by aggregate level simulations. From these we have derived subclasses that refine details required by progressively higher fidelity or lower aggregation models.

Exercise Area

At the most abstract level, an Exercise Area with default atmospheric conditions is defined (figure 3). The Exercise Area identifies the limits within which all operations take place and is made up of one or more Geographic Divisions. Only one Exercise Area object can be created for each exercise. The Exercise Area class contains attributes and services that apply to all the Geographic Divisions or across Geographic Divisions.

Geographic Division

The Geographic Division class contains the geographic details of a subdivision within the

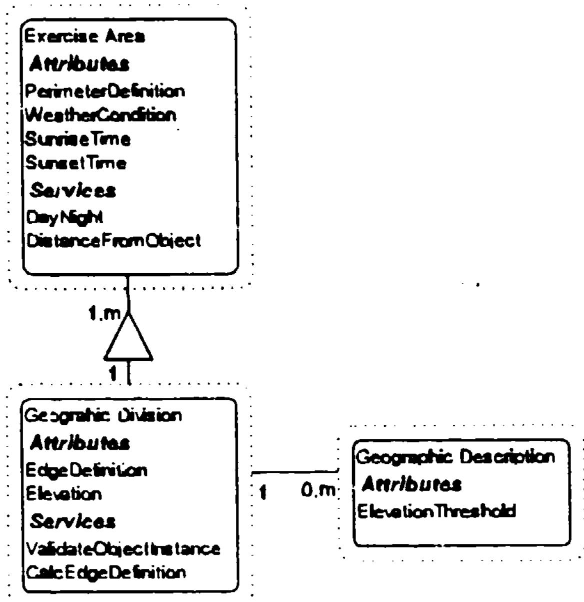


Figure 3. Exercise Area

Exercise Area. The triangle in the figure expresses a whole-part relationship between the Exercise Area and its Geographic Divisions and no inheritance takes place. Since the Exercise Area does not contain any geographic information, the complete area must be described by a series of Geographic Divisions, as indicated by the 1 to many (1,m) Geographic Division parts of the Exercise Area. Currently, we are describing Geographic Divisions as irregular polygons enclosing points of homogeneous data, but a more traditional regular polygon grid can easily be used and may simplify several operations.

The number of Geographic Divisions used to define a given Exercise Area varies with the purpose and level of aggregation of the exercise. The determining factor is the definition of homogeneous data. For instance, the elevation threshold in a homogeneous Geographic Division defined for a deep strike air exercise might be 100 meters while the same Exercise Area described for a battalion level ground exercise might require many homogeneous Land Geographic Divisions each containing no more than 1 meter terrain variation. A single Geographic Division can be coextensive with the Exercise Area as is currently true for most open-ocean naval exercises. At the other end of the spectrum, the importance of terrain variations in land operations generally dictates the use of multiple Geographic Divisions in even the smallest exercise.

The meaning of homogeneous for each data field that describes the Geographic Division and its subclasses is defined in the Geographic Description objects. The threshold for each field is uniform across the Exercise Area, i.e., only one Geographic Description object exists no matter how many Geographic Division objects are created. The value of the thresholds can be changed for each exercise.

Because the data required to describe a Geographic Division differs greatly between Land and Sea, the Geographic Division class is usually specialized to a Sea subclass (figure 4) or a Land subclass (figure 5). Each of these subclasses inherits the attributes and services of the Geographic Division class. Specialization and inheritance are indicated by the half-circle node.

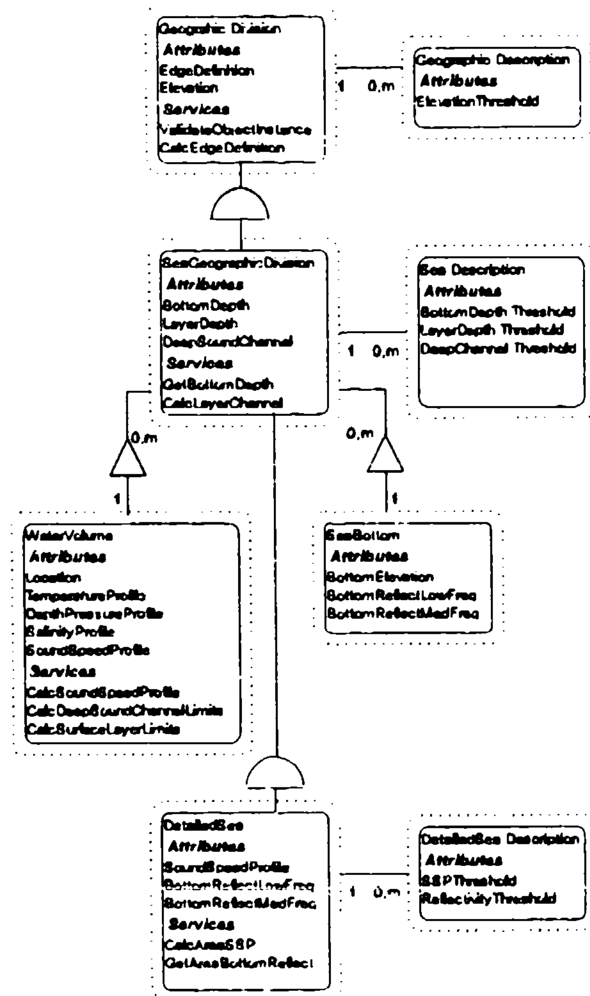


Figure 4. Sea Geographic Division

Sea Geographic Division. The Sea Geographic Division class contains data that influences both trafficability and sound transmission. Further specializations of the class allow more detailed exposition of the data.

At the less specialized level, the Sea is described simply by bottom depth, acoustic layer depth, and deep sound channel limits. This data may be provided *ad hoc*, as it might be in a highly aggregated game, or may be derived from a detailed data base using the thresholds supplied in the Sea Description. The Detailed Sea divisions are more appropriate for tactical situations and are based on large data bases of information.

Land Geographic Division. The Land Geographic Division class describes the terrain trafficability and, if necessary, elevation. If both air and land interactions are to take place, the Elevation refines the Elevation defined for the Geographic Division. Trafficability is currently described as an enumerated value. A moving entity determines the effect of the trafficability on its own movement Because trafficability can change as a result of weather or human activity, the Land Geographic Division has the capability to recalculate its trafficability and inherits the ability to redefine its own limits if the trafficability is changed unevenly. Redefinition of limits requires, of course, the initiation of similar activities at surrounding Land objects and the possible creation of new Geographic Division objects.

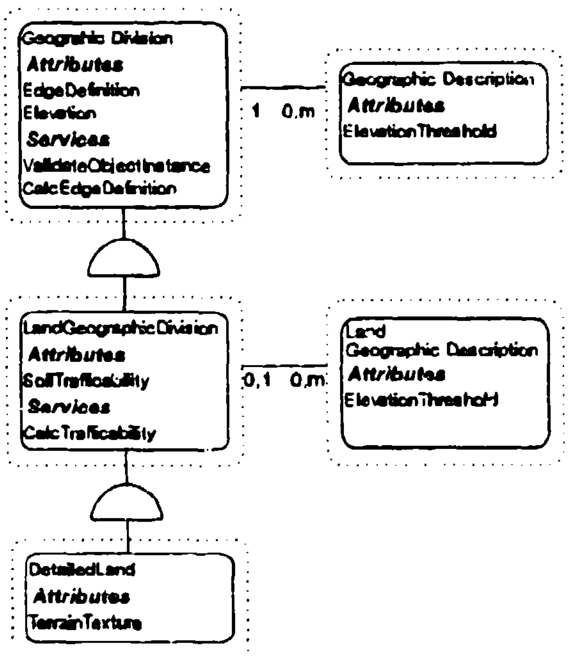


Figure 5. Land Geographic Area

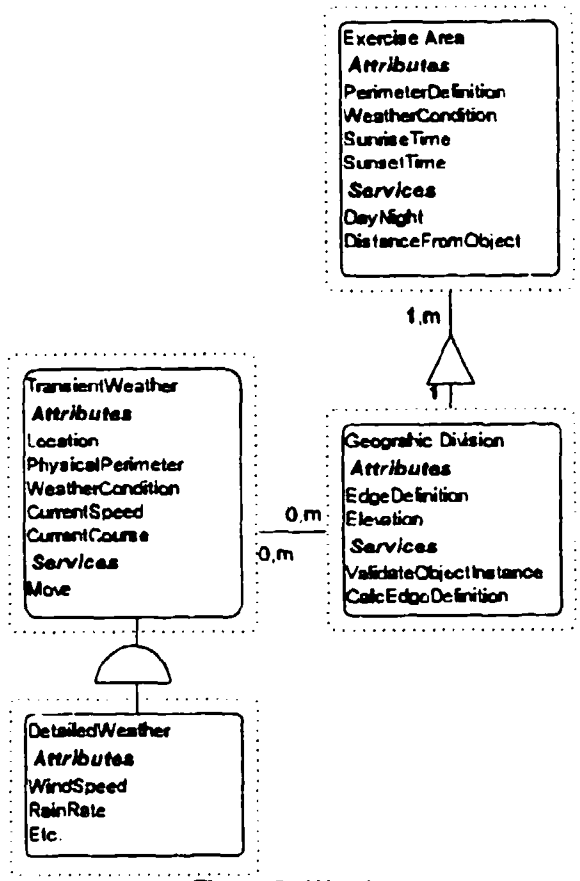


Figure 6. Weather

The Land Geographic Division class as defined can be used for aggregate simulations. Application to unit level simulators that depend heavily on computer imaging of the terrain requires a specialization that incorporates imaging data. This data would include but not be limited to the Terrain Texture currently included in the Detailed Land class.

Atmosphere

Atmospheric conditions affect operations on land and at sea, as well as in the air. The most common atmospheric values currently modeled are day-night and a general weather condition across the area. Because at the highest level of aggregation these may be the only atmospheric

conditions needed, dawn/dusk and a generic WeatherCondition, which is an enumerated value, are part of the definition of the Exercise Area. See figure 6. Sunrise and sunset, rather than day/night, are specified in order to allow a varying day length and to support the use of algorithms that calculate shadow based on the angle of the sun and the variation in daylight across a wide area. These variables are most useful for air operations.

The default weather condition is applicable unless a more detailed, local weather condition is defined by a TransientWeather object. Because TransientWeather is a moving entity that overlays the Exercise Area and may cause changes in the Geographic Divisions, it periodically informs the Geographic Division objects of its position, course, and speed and the Geographic Division determines the effect of the weather on its attributes.

Detailed Weather is being defined to allow the breakdown of the enumerated weather condition into its component parts for use in algorithms that use more detailed data.

Events

As described in figure 7, Any Object, such as a cultural feature (e.g., buildings, landing strips, and roads) or force unit, is located within a Geographic Division. Each Any Object may also be located within a TransientWeather area. Events, such as movement and detection, involving an object are conditioned by these relationships. Which algorithms are used to calculate events depends on the data available from the object or objects involved in the event and from their environment. The object with the least detailed level of data controls algorithm selection.

If we examine a detection event involving a submarine, we can see how such an event might be modeled in an environment described at different levels of abstraction. The selection of which model to use takes place within the Interaction Event, in this case a Detection Event, based on the algorithms it has available and on the data it can access from the interacting objects and from the environment. For simplicity, we assume that all required data is available from the detecting and detected object.

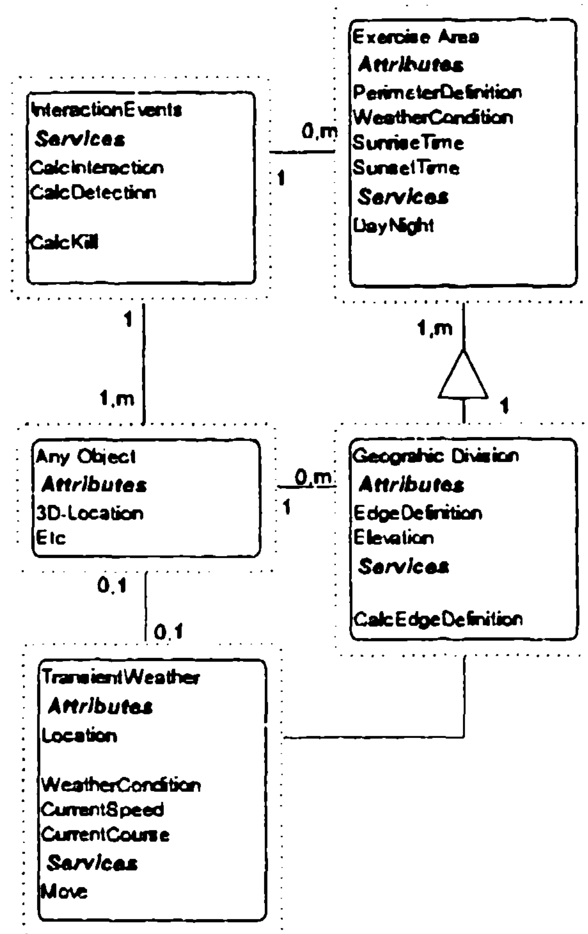


Figure 7. Events

Single Geographic Division. No environmental data specifically related to submarine detection and interaction is available. If a submarine interaction were modeled in this environment, detection/engagement/kill probabilities would be assigned based on the attributes of the interacting objects and, possibly, the weather.

Single SeaGeographic Division. The positions of the interacting objects in the water column influence the detection probabilities. Engagement and kill probabilities are based on the interacting objects.

Multiple SeaGeographic Divisions. Both water columns and fronts can be taken into account.

Single DetailedSea. A detailed range-independent propagation loss model can be used to determine signal loss and, if applicable, reverberation. This data then becomes input into a detection algorithm or screen display.

Multiple DetailedSeas. Range-dependent propagation loss models can be used. Whether the multiple divisions are grids or polygons will affect model input.

The same data base of environmental data can be used to build the environment at different levels of abstraction. Detection Events occurring at these varying levels can be modeled to accommodate the level of data available and the level of detail required. The detecting and detected object do not need to know the level of abstraction nor does the environment need to be aware of the "Any Object" description. The requirements of the interaction are hidden in the InteractionEvent, which in our example is specifically a Detection Event.

Conclusion

Distributed simulations are still in the early phases of evolution. Joint operation of disparate subsystems seems possible if the level of aggregation difference is not too great. Designing environmental models that are useful across a range of aggregation and responsive to the needs of simulators with differing feedback requirements remains challenging. Substantial effort will be needed to tune models to meet the desired level of output consistency.

Object oriented methodologies point towards potential solutions. The hierarchical architecture described above provides a layered approach to development of consistent models that can operate across a range of levels of aggregation in the overall simulation. New models designed to extend the range of validity can be developed and inserted in a straightforward way without disrupting existing models.

The object oriented approach has been shown to provide substantial benefits in modularity, upgradeability, portability, and information hiding. The approach explored here also appears to hold promise for more general methods to decouple models, such as environment, from the unique components in the simulator. Decoupling and standardizing environmental models and object hierarchies can result in more rapid and cost effective development of future simulator subsystems built to interact on the distributed net.

Endnotes

1. Acronyms used in figure 1:
   - JTLS: Joint Theater Level Simulation
   - NWGS: Naval War Gaming System
   - NAVTAG: Naval Tactical Game
   - TASWIT: Tactical Advanced Simulated Warfare Integrated Trainer
   - DEFTT: Decision Evaluation Facility for Tactical Team
   - SAAT: SURTASS Acoustic Analysis Trainer
   - TNOTS: U.S. Naval Academy Tactics, Navigation, and Operations Training System
   - SWATT: Surface Warfare Advanced Training Technology
   - AAT: Acoustic Analysis Trainer
   - PORTS: Portable Electronic Threat Simulator
   - GSS: Generalized Simulation-Stimulation
   - TSOT: TRIDENT Sonar On-Board Trainer
   - CCTT: AN/BSY-1 Combat Control Team Trainer
   - ACTS: AEGIS Combat Training System

2. The notation used is described in *Object-Oriented Analysis* (2nd ed.) Peter Coad and Edward Yourdon, 1991.

DISTRIBUTED SIMULATION: DOES SIMULATION INTEROPERABILITY NEED AN ENVIRONMENT SERVER?

Gary M. Kamsickas
Boeing Defense and Space Group
Simulation and Training Systems
Huntsville, Alabama

ABSTRACT

Future Distributed Interactive Simulation (DIS) implementations will be significantly impacted based upon the resolution of issues relating to a distributed versus a central computer generated environment .

DIS, developed by the University of Central Florida Institute for Simulation and Training (UCF/IST) and funded by the Program Manager for Training Devices (PMTRADE), is based on the Defense Advanced Research Projects Agency (DARPA) developed Simulation Network (SIMNET) technology. DIS is an architectural approach where large scale, multi player simulation is distributed across independent and self sufficient computers instead of one central computer. DIS implementations are used to train individuals in coordinated team tactics and support weapon system evaluation through test/prototype developmental systems in realistic simulated combat scenarios. One of the key concepts behind the DIS architecture is the autonomy of each individual simulation. This implies that each simulation entity is responsible for maintaining a realistic, true representation of the environment external to itself. Several problems arise when large numbers of simulation entities of different fidelities and designs interoperate within a DIS architecture based exercise. Since there is no central source of "ground truth" for the environment, each simulation provides a specific internal representation of the environment. Because each simulation device operates within this internal environment , each player could potentially have a dramatically different representation of the simulated external environment. The result is a situation where a "fair fight" is not possible among the players. It has been suggested in the simulation interoperability technical community that a central environment, or "Environment Server" approach, if implemented, could reduce or eliminate this problem along with several other issues related to the commonality of the simulated environment. This approach seemingly violates the autonomy goal of the DIS architecture. This paper discusses several key issues and the relative advantages and disadvantages between the distributed and centralized environment approaches. The resulting impact to future DIS implementations of each approach is assessed. A hybrid approach that takes advantage of the strengths of each approach is presented.

ABOUT THE AUTHOR

Mr. Gary M. Kamsickas is a Software/ Systems Engineer with the Simulation and Training Systems organization of the Boeing Defense and Space Group in Huntsville, Alabama. He has been responsible for software design, code, test , Systems Engineering and integration on several Boeing simulator projects, including the Ada Simulator Validation Program (ASVP) and the Modular Simulator Design Program (Mod Sim). Mr. Kamsickas holds a Bachelor of Science degree in Electrical Engineering from Michigan Technological University, Houghton, Michigan.

DISTRIBUTED SIMULATION: DOES SIMULATION INTEROPERABILITY NEED AN ENVIRONMENT SERVER?

Gary M. Kamsickas
Boeing Defense and Space Group
Simulation and Training Systems
Huntsville, Alabama

INTRODUCTION

For simulated entities to effectively and realistically participate in the same exercise, they must have access to the same simulated environment information. Different types of information about the environment are necessary in order to make the exercises as realistic as possible. This information may include changes in terrain, weather, tactical smoke, ambient illumination, magnetic variation, earth model definition, electromagnetic effects, etc. The types of environment information required for an exercise will vary depending upon the types of entities that are interacting and the goals of the exercise. Distributed Interactive Simulation (DIS), a distributed architectural approach where large scale, multi player simulation is distributed across independent and self sufficient computers instead of one central computer, is currently the standard for interoperability for simulation devices. DIS exercises are used for training individuals in coordinated team tactics and to support weapon system evaluation through the testing of prototype/developmental systems in realistic combat scenarios. The DIS architecture is based on the autonomy of each simulation node. This implies that each node is responsible for maintaining it's own perception of the simulated environment in which the entities that it controls operate. There is no central source for the correlation of these environments or the resolution of conflicts between entities. It is imperative to future DIS implementations that a method be developed that provides for this correlation between environments. A suggested approach to resolve this problem is a central or common environment, also referred to as an 'environment server'. This approach could be used to replace or supplement the distributed environment approach now used in DIS. Several key technical issues must be studied in assessing the relative advantages and disadvantages between the distributed and central environment approaches before making a design decision. This paper discusses these issues and provides a potential solution to this problem.

DISTRIBUTED INTERACTIVE SIMULATION OVERVIEW

The primary purpose of DIS is to create synthetic, virtual representations of warfare environments by systematically connecting and establishing a means of communication between separate subcomponents of simulation which have the capacity of residing at distributed, multiple locations. This allows DIS implementations to perform three primary functions; training of individuals and teams, combat/tactics/weapon system development/experimentation and large scale simulation. The separate simulation subcomponents may consist of dissimilar devices including individual man-in-the-loop simulators, semi automated forces or computer generated simulations, groups of simulation devices or the actual weapon system. The simulation subcomponents (nodes) communicate using predefined and mutually agreed upon data packets called Protocol Data Units (PDUs). The PDUs are communicated via local area networks, wide area networks and other forms of communication media depending on the subcomponent's geographical location. There are several basic supporting concepts that have governed the the development of the DIS architecture. They include:

1. There is no central computer which coordinates/schedules events or resolves conflicts between simulation entities. The distributed simulation approach requires each simulation node's host computer to simulate the state of each entity controlled by the node and communicate this information to the other simulation nodes in the exercise.

2. The DIS simulation nodes are totally autonomous and responsible for maintaining the

state of the entities simulated by the node and their respective operating environment. In a DIS exercise each simulation node communicates the state of the entities it controls (location, orientation, velocity, articulated parts position, etc.) to other simulations on the network. The receiving simulation is responsible for taking this 'ground truth' data and calculating whether the entities represented by the sending simulation are detectable by visual or electronic means. This perceived state of the sending simulation is then displayed to the receiving simulation as required.

3. As stated earlier, all simulation nodes communicate 'Ground Truth' data through a predefined, mutually agreed upon, standard protocol. This protocol is documented for all DIS users/players in the form of PDUs. The PDUs are communicated based on a set of data transmission rules which are part of the interoperabilty standard.

4. Receiving nodes are responsible for determining what is perceived based on PDU information.

5. Simulation nodes communicate only "changes" in the state of the entities they control.

6. Dead reckoning or predictor algorithms are used to reduce communications processing and smooth the representation of entities between state updates. Dead reckoning reduces the frequency of transmission for entity state change information.

DISTRIBUTED ENVIRONMENT CONCEPT

The distributed environment concept assumes that each simulation node is autonomous and responsible for maintaining its own view or replication of the environment in which it operates. This approach follows the basic concepts that the DIS architecture and protocols are based upon. The major contention with the distributed environment approach is that there is no assurance of complete correlation among the distributed environments in each simulation node. This is due to differences in simulation modeling techniques, equipment capabilities, databases, etc. internal to each simulation node. The result is a potentially unrealistic or unfair interaction between entities. This could taint the results of the exercise or reduce the effectiveness of the training. There is also a certain amount of redundancy of effort

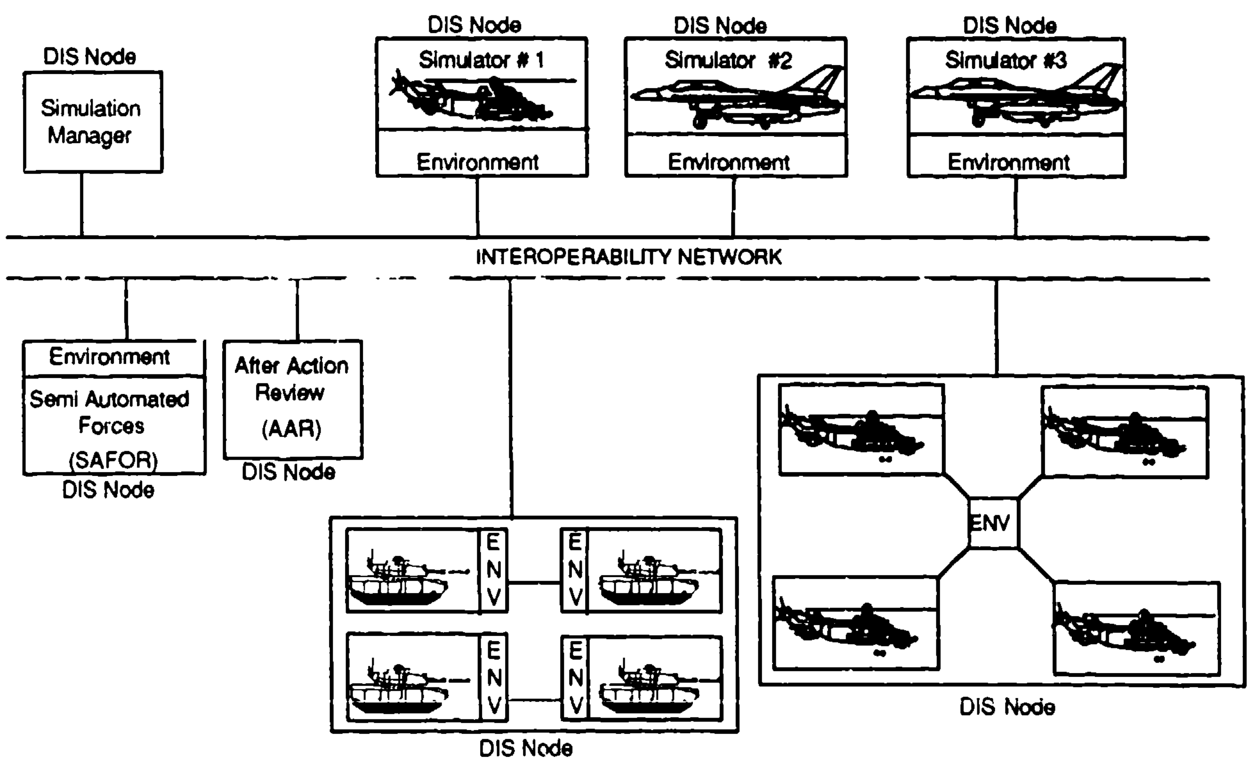


Figure 1 Distributed Environment Concept

associated with creating and managing an environment for every simulation node. Figure 1 illustrates the distributed environment concept architecture which also complies with the DIS standard.

CENTRAL ENVIRONMENT CONCEPT

The central environment concept assumes that there is one central environment or environment 'server' which supports all nodes in an exercise. The main point of contention with the central or common environment approach is that there must be a central computer, or a computer on one of the simulation nodes. that performs the role or functions of the central environment. This conflicts with several of the basic premises supporting the DIS architecture. The environment server would maintain a correlated, common view of ground truth for the environment for all entities, thus eliminating correlation anomalies. Another benefit of the environment server approach occurs in large scale simulations. In these instances each simulation node would not be required to process environmental data packets and changes in the environment, thus reducing the processing capacity required for each node. Large scale exercises could require a significant amount of processing power for each node. This could virtually eliminate low end simulators from participating in the exercise. Figure 2 illustrates the central environment concept architecture.

TECHNICAL ISSUES

There are several technical issues that should be considered when determining the best solution to the environment issue. Each of these issues has a distinct impact on the approach that could be used for the DIS environment implementation. The resolution of some issues has an impact on other issues. Where this occurs the author has attempted identify this connectivity.

Interoperability of Varying Fidelity Simulations and Existing Simulations

One of the goals of the DIS concept is to allow any simulation asset, including new or existing devices, to play or operate in a DIS exercise. This would include both high fidelity and low fidelity systems. A problem arises when combining simulations of

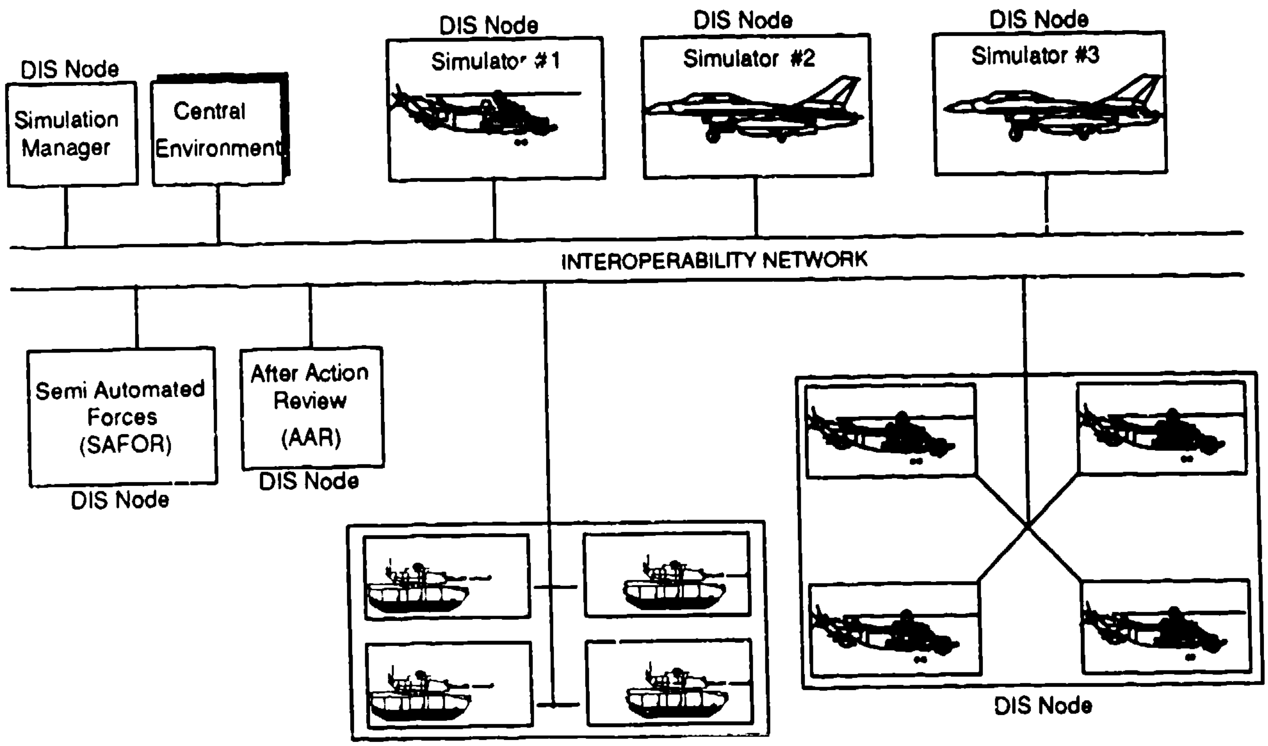


Figure 2 Central Environment Concept

varying fidelities. In some cases, the higher fidelity simulation may have a distinct advantage over the lower fidelity device thus impairing the capability of a "fair fight" between the entities. For example, a higher fidelity sensor or visual simulation may allow for more rapid detection of a target over a lower fidelity simulation. In this case the higher fidelity system would have a distinct advantage. The fidelity difference could also work to the advantage of the lower fidelity simulation in other situations. A lower fidelity simulation may not consider atmospheric effects that would hinder detection, therefore allowing quicker detection on the part of the lower fidelity simulator. A central environment could help this situation by providing a more "leveled" interoperability environment. Every device in the exercise could use the central environment. If the central environment was aware of the fidelity limitations of each player, it could perform some sort of "leveling" to allow a fairer fight, or more realistic data collection if the exercise was an experiment. Another advantage of the common environment is that the lower fidelity simulations get access to the higher fidelity common environment, thus improving the fidelity of the device while maintaining its low cost.

There is general consensus that an exercise must be "fair" or "level" for all players, but a common environment does not solve this problem. The method in which the simulations use the common environment must also be the same to approach a fair fight situation. Providing high fidelity environment data to a device that makes modeling assumptions that do not use the data is a waste of time. There are some cases where even identical manned simulations cannot have a fair fight due to visual system interoperability issues. Different types of simulation devices also have different environment requirements for fidelity. For example, a tank or ground vehicle is concerned with trafficability and shadows for concealment. Whereas an airborne vehicle has no need for fidelity in these areas. The common environment would have to be all things to all users. This is a tremendous task. The environment server can help the "fair fight" issue, but it is impossible to determine how much and at what cost. The interoperability of varying fidelity simulations is a very complex issue and beyond the scope of this paper. The environment approach used to achieve a fair fight is a small portion of the solution to this issue.

Visual System Interoperability

One of the major interoperability shortfalls is in the area of visual systems in manned simulation devices. The seemingly endless variety of image generators, display systems and techniques for rendering images can yield a plethora of visual representations of the environment. Visual system interoperability is also probably one of the largest contributors to the "fair fight" issue between simulations. Once again, a common environment might solve some of these problems but not all of them. One example is visual display systems. Each manned simulator will have a visual display system with a specific field of view. The simulator with a larger field of view would have an advantage over a simulator with a smaller field of view because it would have a better chance at target detection (assuming both systems have the same resolution). Neither environment approach would improve this situation. The image generation system would also have a significant impact on the capability to render a common view of the environment due to different image generation techniques. One of these techniques is the rendering of higher detailed images based on eyepoint or distance from the participant. To get more image generation power, the image generator produces high detail images only where the participant is looking or only for a certain visual range. In this case even identical image generators may not yield leveled interoperability conditions in some situations.

A common environment may improve the issue of visual system interoperability but only to a very limited extent. A common environment could help level the playing field. The extent of this "leveling" over the distributed environment approach would be small and highly dependent on the implementation of the common environment's design.

Common Databases

The common database issue is concerned with different simulations having different database representations of the simulated world. It is obvious that without proper correlation and commonality between databases, anomalies will exist that remove the realism of the simulation (tanks floating above the ground) or reduce the ability for a fair fight between players (differing line of sight calculations). The databases include

visual, terrain, navigation/communication, and sensor databases. It stands to reason that independent development of environment databases will yield different results. Database features such as trees, buildings, and terrain contours will not be identical in each database. A solution to this is to have a common environment database. This would require each simulation to either use the common database or do a translation to move from the common format to the simulation's specific image generator format. The government's Project 2851 is moving in the right direction for this initiative. However, there will be a cost to convert older, existing simulations to a new database. The solution for this issue is not an environment server but a common environment database baseline that each simulation can use to create its own database and allow for distributed operation. A predefined set of interoperability databases could be developed to be used by DIS exercise participants. This would improve the correlation between distributed environments. This approach would not preclude the implementation of a common environment approach. The common environment approach could use the common environment database.

Dynamic Terrain and Cultural Features

During the course of a DIS exercise, changes may occur in the terrain or cultural features such as bridges or buildings. These changes are due to interaction between the entities and the environment. Munitions may be fired which impact the ground creating craters or impact bridges or buildings destroying cultural features. Entities may also create dynamic changes in the terrain by digging ditches or creating embankments for tactical reasons. These changes must be communicated to the exercise participants to ensure that the environment remains common to each player. There is currently no method for this in the DIS standard. Proponents of the common environment suggest that communicating this information to a single source and that single source interpreting the data is the most efficient solution. The environment server would record all changes to the terrain/cultures and inform players as required. The distributed environment approach does not work well for dynamic terrain and cultural features. This is due to the concept of late player entry. How late player entry affects the environment issue is described in the following paragraph.

Late Player Entry

DIS exercises must allow for the entry or restart of players into an exercise that is already in progress. This presents a problem in the area of dynamic changes to the environment such as dynamic terrain, changes to cultural features, weather/atmospheric changes, and long term dynamic effects such as high winds changing the wave height on the sea or long periods of rain causing the ground to become muddy. When a new player enters an exercise, it must be informed of these dynamic changes to have a correct representation of the environment. The problem with many of these dynamic changes, such as a crater being formed when a munition detonates, is that they are treated as a single event. If a new player enters the exercise after the event has occurred, it's view of the environment does not contain the results of the dynamic event. An environment server could keep track of and record all of these changes and transmit them to the new player during initialization. The environment could also incorporate them into the central environment database and provide the database to the new player. The latter would require a significant amount of network bandwidth. There are two methods to handle this problem in a distributed environment. In the first method the simulation manager could collect and record the dynamic environment changes and provide them to the new player when it enters the exercise in the same manner as initialization data is handled. This is similar to the environment server except that the simulation manager is performing one of the environment server's functions. The second method would be to make each dynamic change an entity. The changes would then be transmitted into the exercise using the same rules as the entity state PDU. Some of the dynamic features, such as smoke clouds and cultural features, have already been used as entities. The drawback to this approach is that entity state data would be continually transmitted for entities that virtually never change, such as craters, just on the chance that a new player would enter the exercise. Depending on the exercise this could be very wasteful in terms of network bandwidth.

Entity Handover and Entity Reconstitution

One of the identified needs for the central environment is to handle long term effects of

entities on the environment. Two examples have been used frequently in past DIS workshops. The first is that of a high speed jet which drops chaff at a certain geographic location and then leaves the location for the remainder of the exercise. The chaff cloud continues to float in the atmosphere for several hours causing its effects on other entities. Meanwhile, the entity which dropped the chaff has completed it's mission and is no longer part of the exercise. The entity's host computer must remain on line to simulate the chaff cloud entity until it totally disperses or until the end of the exercise, thus wasting computational resources. This same example could be applied to land and water mines, sonobuoys, etc. The second example deals with a tank or other entity being destroyed in a battle. After destruction, the entity remains in the exercise as a burning, smoking pile of rubble. Again the entity's host computer must remain on line to simulate the destroyed entity until the end of the exercise. The Entity Handover PDU could be used to handover the simulation of these residual effects to a central environment server. Then the host computers that created the effects could be removed from the network or used to reconstitute new entities as required by the exercise.

This rationale is a weak excuse for an environment server. It only relocates a simulation to another host and introduces additional problems into the system. One problem with this approach is that it can potentially reassign a virtually unknown amount of processing to a potentially overloaded computational asset. The environment server would have to be sized to accommodate the worst case processing load. This was one of the reasons for the underlying distributed nature of the DIS architecture. The environment server must also be capable of performing the modeling of a potentially endless variety of entities that could be passed to it. The solution to this problem is simple. Host computers should be sized according to the mission of the entity or entities that they support. If the host computer serves a jet aircraft capable of dropping fifty rounds of chaff, then the host should be sized to support that requirement. As for the reconstitution of a destroyed entity by a host computer and the resulting maintenance of a burning mass representing the destroyed entity, this should not be a problem. The burning mass is simply another entity that is no different than a fired missile. An entity representing burning mass should not require a great deal of processing resources. It should not have any rapid changes and therefore only require an entity state PDU transmission every five seconds. In short, the environment server should not be used as an entity servicing device for other hosts.

Reliability and Maintenance

Reliability and maintenance must be considered in making a decision between the two environment implementation approaches. If a distributed environment approach is used and a simulation node fails, it can be removed from the exercise and replaced with another node if available. In most cases the exercise could be held as planned or conducted without the failed node. If a central environment approach is used and the central environment server fails, the entire exercise must be cancelled or rescheduled until repair of the environment server. The environment server becomes a single point of failure for the system. In addition to reliability, we must also consider maintenance. Distributed environment nodes are responsible for maintaining their own equipment and its configuration as part of their respective contracts. In an environment server approach, someone must be responsible for maintaining the central environment hardware and software. This would include configuration control of the hardware, software, users guides and other documentation, resolution of user interface problems, scheduling of resources, and distribution of changes and user data. This ongoing cost of maintenance and the reliability risk make the central environment approach less desirable.

Environment Reuse

Another issue that should be considered in the analysis of central versus distributed environment is the potential for reuse. Most of todays new software development initiatives are focusing on the reuse of software from one device or program to the next to save substantial redevelopment costs. It would appear that the central environment is a clear winner in this area. The central environment would be used by everyone in the exercise thus saving everyone (except for the initial cost of developing the central environment) the expense of developing and maintaining environment software for each simulation node. In reality this would probably not be the case. Many simulations will be developed to operate as stand-alone entities in addition to there use as players in a DIS environment. Except for very low cost

simulations or semi automated forces, it would not be cost effective to develop an entire multi million dollar simulation device only to have it be dependant on using an environment controlled by someone else. This is again affected by the maintenance and reliability issue. If the simulation does not have its own environment model, then it would be rendered inoperable as an individual trainer and as a team trainer if the central environment was unavailable. The central environment may also not be capable of supporting a specific mission that the individual device wishes to train due to the current version of the environment or of the device. This could cause a significant impact to an individual organization's training plans and schedules. We must also consider that older, existing simulation devices will already have their own internal environment from their initial development. These internal environments would have to be modified to "switch off" when in a DIS exercise.

Does this mean that there is no hope for environment reuse in DIS? Of course not. A robust environment simulation that is compatible with the DIS interface and individual simulations could easily be developed and reused on individual devices. This environment simulation could be applicable to both existing and new simulation developments. An example of this approach can be found in the Modular Simulator System generic specifications. An environment segment has been allocated as a major portion of the Modular Simulator architecture. This segment serves as a single environment focal point for the simulation regardless of whether the simulation is operating as a stand-alone, autonomous device or as player in a DIS or other multiple entity exercise.

Real World Devices

A future goal of the DIS concept is to allow real world devices, manned and unmanned, to play in exercises with simulated entities. How will these real world devices deal with the environment, either common or distributed? The real world entity has a real environment not a simulated environment. This may be a case where the central environment could be very useful if the real world device was capable of "disconnecting" from its real environment. The real device could use the central environment and all of its associated services without having to develop its own environment. The requirement for real world devices to operate in a DIS exercise would also support the need for a reusable environment simulation for a distributed environment approach. The requirements for the interoperability of real world devices in a DIS exercise have not been fully defined. There are a large number of unique implementation details that are specific to this issue that will affect the real world device to DIS interface that have not been investigated. It is impossible at this time to recommend an environment approach that addresses this issue.

THE ENVIRONMENT PROTOCOL DATA UNIT

The current DIS environment working groups are in the process of developing an Environment PDU that will be used to communicate state changes of entities/features in the environment. This PDU was not complete as of the writing of this paper. The PDU assumes that all exercise participants will start from a predefined initial environment state. The mechanics of how this predefined state are provided to the players, its format, and method of implementation are still unresolved. The environment PDU is still in a draft state and in review/modification by the various DIS environment subgroups. It is not currently part of the DIS draft standard. The environment PDU appears to be flexible enough to support either of the central or common environment implementations.

SUMMARY/CONCLUSIONS

Considering the data presented, it would seem that each environment approach has some merit. Operating in a DIS exercise involves technical issues, logistics, politics, contractual relationships, real world hardware, semi-automated forces, existing simulation assets and a whole realm of other variables. Neither of the environment approaches can effectively address all of these issues. As with most solutions there is always a compromise. A hybrid environment approach is provided in Figure 3 that illustrates a compromise between the two approaches. In this approach there is no environment server or common environment connected to the interoperability network. The hybrid approach is based on certain functionalities being provided by the DIS simulation manager. Since the simulation manager is currently undefined in the standard, it is fair to assume that these functions could be incorporated into its specification. At the beginning of an

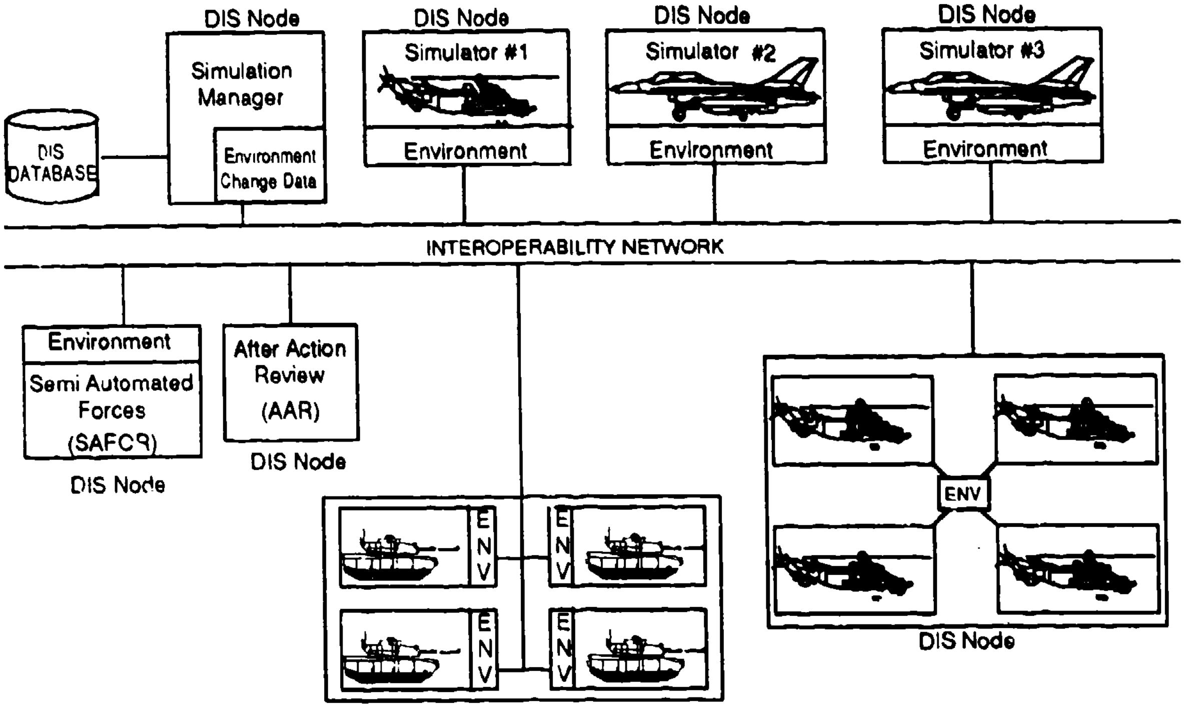


Figure 3 Hybrid Distributed Environment Concept

exercise the simulation manager identifies to all players the common environment database to be used for the exercise from a repository of existing DIS databases. In most cases it would not be feasible to download the actual database to each node via the network. In addition to requiring a large amount of network bandwidth to transmit the data, most nodes will probably need to do a translation or some other sort of preprocessing to create a database that the node's hardware can understand prior to running the exercise. The databases could be predistributed and preprocessed by the nodes. The database to be used in the exercise could then be communicated by a code. For example, "Database 3, Version 2.1". This solves the common database issue and improves the interoperability between visual systems and devices of varying fidelity. It also promotes environment reuse among the nodes. For the remainder of the exercise, the simulation manager maintains a record of environment change data. Environment change data will be communicated during the exercise using the Environment PDU The simulation manager will provide the environment change data to new/late players as they join the exercise using the same PDUs defined for initialization data. To reduce the amount of environment change data required in a DIS implementation, future DIS working groups should identify specific environment features that could best be represented as entities. These environment features could then be handled with the existing Entity State PDU. This method of handling environment change information solves the problems associated with the dynamic terrain and late player entry issues.

This solution maintains the basic DIS concept of node autonomy at the DIS interface level. It does not preclude the concept that an environment server could be used internal to a DIS node to provide a common environment to a group of entities within the node. This approach will cause some amount of maintenance for the common databases, but this should be minimal. The basic concepts behind this approach are sound and support the basic goals and design concepts behind the development of the DIS architecture. Some minor modifications to the details of operation may be required to make use of existing DIS PDUs or to align with the existing functions planned for the simulation manager.

REFERENCES

IST-CR-93-15 Proposed IEEE Standard Draft, Standard for Information Technology - Protocols for Distributed Interactive Simulation Applications Version 2.0 Third Draft

WEATHER ENVIRONMENT SIMULATION TECHNOLOGY

Bruce C. Montag
Southwest Research Institute
San Antonio, Texas

ABSTRACT

This paper presents the results of an internal research program at Southwest Research Institute for the development of a weather simulation and modeling approach for training and simulation applications. This weather simulation system approach, known as Weather Environment Simulation Technology (WEST), provides the means to correlate and synchronize all weather-related cues presented to the student. The approach provides for direct correlation between out-the-window visual weather scenes, weather-processing sensors and avionics displays, and vehicle handling qualities through the use of a unified meteorological database that has been reformatted specifically for real-time simulation. By ensuring dynamic weather cue correlation across all simulator subsystems, this technique enables simulator instruction in weather-related procedures to be highly transferable to mission-oriented situations. This research effort demonstrated a method for processing weather data in real time for generation of out-the-window weather imagery that correlates directly with airframe dynamic effects. The model architecture also supports sensor simulations and generation of cues on operator displays and controls. Since the weather model is driven by gridded-field, digital meteorological data, students can learn and practice weather-related skills within a realistic, synthetic weather environment as produced by a WEST-compatible simulator.

ABOUT THE AUTHOR

Bruce Montag is manager of the System Simulation and Modeling Section in the Training Systems and Simulators Department at Southwest Research Institute. He directs the operations of the Visual Simulation Lab at SwRI and has over ten years of experience in simulation and modeling. As principal investigator for the WEST research project, he originated and implemented a unique means for visually simulating atmospheric weather effects. Prior to joining SwRI, he led the tactical avionics and weapon simulation efforts for the LANTIRN and F-16C simulator programs at Link Flight Simulation. He holds a BS in Mechanical Engineering from the University of Texas at Austin and an MBA from the University of Houston at Clear Lake.

WEATHER ENVIRONMENT SIMULATION TECHNOLOGY

Bruce C. Montag
Southwest Research Institute
San Antonio, Texas

INTRODUCTION

Weather plays a key role in the planning and performance of joint operations involving the tactical employment of sensors and weapon systems. Yet weapon system trainers (WSTs) and networked distributed interactive simulation (DIS)-compatible simulators are currently limited in their ability to provide a realistic portrayal of weather conditions that are applicable to these types of mission operations.

Needs and Requirements

Sources of sensed atmospheric data have expanded rapidly within the last several years, including ground-based doppler weather radars and satellite-based systems that sense and record large-scale weather conditions in the form of time-varying 3-D (space plus time) gridded-field, digital data. Although these systems are producing voluminous amounts of data for forecasting and analysis purposes, methods for applying this data effectively within simulators for training, mission planning, and mission rehearsal are needed.

Approach

The WEST approach outlines an efficient way for integrating digital weather data, structured in a gridded Cartesian format, into manned simulation systems. Both uniform and non-uniform grid formats may be accommodated. WEST provides the means for simulators to manipulate atmospheric data rapidly and transform physical weather parameters into simulator display cues that are correlatable across individual simulator subsystems. Weather data parameters accommodated within WEST include wind direction and magnitude, liquid water content, temperature, pressure, radar reflectivity, and water content type (rain, snow, ice). Digital atmospheric data sets currently supported by the approach include the Joint Airport Weather Studies (JAWS) database from the National Center for Atmospheric Research (NCAR) and the Terminal Area Simulation System (TASS) meteorological weather model developed by the NASA Langley Research Center for windshear research.

With the WEST approach, digital atmospheric source data is reformatted and preprocessed to form a unified weather database capable of supporting each simulator subsystem. During simulator operation, a weather "generator" accesses the unified database and provides a sensor-specific formatted list of weather elements located within each subsystem's field of regard. The subsystems incorporate the weather data into sensor processing operations and transform the sensed digital weather data into simulator display cues.

Applications

WEST is designed to support weapon system training applications, mission rehearsal, flight training, and weather analysis applications. Although work to date has focused on generating visual imagery from gridded weather data sets, the approach is extensible to supporting FLIR simulations, digital radar landmass simulators (DRLMS), and other types of sensor simulations.

BACKGROUND

The atmospheric environment is a common denominator that affects virtually all military operations. Aircraft, ships, tanks, sensors, dismounted infantry, weapons, and communications system, must perform their individual tasks under a wide range of weather conditions, anywhere in the world, at any time of year. Providing the warfighter with the tools to understand, cope with, and take advantage of the weather environment and its effects on mission tasks is a pressing need within the simulation and training community.

Mission Operations Affected by Weather

Specific mission operations that are most affected by spatial and time-varying weather conditions include aircraft take-off/landing, air combat tactics, target acquisition, and weapons delivery, to name a few.

Aircraft Take-Off/Landing. Aircraft are most susceptible to the effects of windshear, heavy rain, and turbulence during the take-off and landing phases of flight. Training aircrews via simulator instruction in how to detect and avoid hazardous meteorological conditions and effects is needed within the military and also within the commercial and general aviation sectors. Adverse atmospheric conditions that impact flight safety include thunderstorms, microburst windshear events, gust front turbulence, wake vortices, and dynamic interface between aircraft and structures, such as those between a rotary wing aircraft and a ship superstructure.

Mission Tactics. Tactical exploitation of area weather conditions can provide an extra edge over an opposing force when planning and performing mission operations. The liquid water content within convective weather masses provides a freely available means for attenuating and masking platform observables (visual, radar, infrared) when the weather mass is placed between the aircraft and the observing sensor. By introducing timely, geospecific gridded weather data into a mission rehearsal or mission planning system, various routes and sensor profiles can be examined beforehand to identify the optimal movement and placement of forces for a given operation. Routing a strike package around the upwind side of a thunderstorm to hide its presence from search radars would be one example of how local weather conditions can be exploited to provide a tactical advantage.

Target Acquisition. The ability of target acquisition sensors such as radars, imaging infrared (IIR) sensors, and electro-optical sensors to discriminate targets is greatly complicated by intervening weather. A similar problem is presented by battlefield smoke. The spatial distribution of water and particulates within the volume of air over the area of operations causes varying amounts of signal attenuation and clutter across each spectral band, making targets and cultural features difficult to distinguish and designate on operator displays. In addition to reducing situational awareness, intervening convective weather and smoke obscuration conditions also can cause seekers/trackers to break lock. This is more of a problem for longer-range sensors and weapons in which clouds or smoke plumes may come into the sensor's field of view while target tracking.

Weapons Delivery. Weapon flyout and delivery accuracy is also affected by adverse weather conditions. Atmospheric wind and density variation have a direct impact on the way in which weapons fly to the target. Both wind variability and density variation affect the aerodynamic forces acting on the weapon, which in turn affect the weapon's flight path.

Current Practice Limitations

Current practices for modeling four-dimensional (space plus time) weather effects for simulator-based training do not yet provide the necessary cues, cue correlation, system performance, or environmental modeling features that are required for truly transferable training to the weapon system for weather-affected operations. Most WSTs in the field today feature separate weather modeling approaches on a per-subsystem basis. Simulator subsystems that model the sensing of weather conditions include digital radar landmass simulators (DRLMS), visual systems, FLIR image generation systems, and airframe/vehicle simulation systems.

Digital Radar Landmass Simulators. State-of-the-art approaches for modeling weather within DRLMS systems involve the use of simple geometric objects (e.g., cylinders, cubes) or digitized two-dimensional (2-D) weather maps for the DRLMS weather database. On systems that use maps as a database source, 2-D map elements are assigned top and bottom altitudes to provide 3-D weather slices. The individual slices are then translated, rotated, and expanded to provide a time-varying radar weather environment. Correlation with other subsystems is attempted by providing the spatial position of active slices back to the host simulation during simulator operations.

Visual Systems. Out-the-window image generators commonly present spatial weather conditions through the use of traditional 3-D

computer graphics techniques that are optimized for real-time image generation involving solid surfaces and/or textured surfaces.

Solid Surfaces. Early flight simulator visual systems used solid-surface polygonal objects to represent cloud formations. These visual cloud representations were created manually using a database modeling system. With the solid-surface approach, clouds are constructed by stretching a tri-mesh-type polygonal surface around the exterior of a cloud. The surface is then colored, shaded, and in some cases assigned a transparency value or material code to provide the cloud's visual appearance. Cloud objects are then grouped together to form weather formations.

The main limitations of solid surfaces for visualizing weather are limited visual realism and the difficulty of correlating visual appearance with digital atmospheric source data. Creating realistic weather effects with solid surfaces requires a lot of hands-on work at the database modeler's station by a highly skilled designer. A mathematical model of weather pattern dynamics must also be developed so that during simulation the individual cloud objects can be dynamically sized, oriented, and positioned. Other significant limitations are that varying resolution models must be developed to accommodate viewing under both far and near distances, and that special visual effects are required to accommodate reduced visibility inside cloud boundaries.

Textured Surfaces. Current-generation flight simulator visual systems use texture to produce photorealistic weather effects. These "texture maps" allow scanned photographic images to be mapped onto polygon surfaces to provide visual weather conditions. The most common practice is to apply photographs of clouds to very large single-polygon "billboards" that are oriented either horizontally (parallel to the ground) or vertically (perpendicular to the ground) to provide the appearance of cloud tops/bottoms or weather formations on the horizon.

FLIR Image Generators. Imaging infrared simulation systems rely upon look-up tables (LUTs) to determine the effect of atmospheric conditions upon the thermo-sensed scene. LUTs are defined for both standard and non-standard day conditions. These tables provide parametric data for altering the color and intensity of scene objects as a function of ambient atmospheric conditions such as temperature and humidity. FLIR simulations do not yet provide the capability to introduce spatial and temporal weather effects into the scene generation process.

Airframe/Vehicle Simulations. Most flight simulators already use some form of gridded-field data structure to model geographically varying pressure, wind, and temperature within the atmosphere on a large scale. This database is accessed during simulator operation by the aircraft equations-of-motion model and the flight instruments simulation. The grid resolution of the pressure, wind, temperature database is on the order of several kilometers, and liquid water content is not included as a database parameter. For these reasons, the pressure, wind, and temperature database used by the host flight simulation is not directly applicable to other simulator subsystems.

Weather Simulation Requirements

An ideal approach for overcoming the limitations of current techniques for modeling weather is to derive simulator cues directly from a unified digital weather database. This provides a much better alternative to the practice of manually creating separate subsystem specific databases (visual, radar, flight) and then attempting to correlate weather cues by "tuning" the individual databases to meet specific training requirements. The WEST approach solves this problem by using a modified simulator architecture that includes additional subsystem interfaces for handling dynamic weather data. Figure 1 illustrates the main idea of the WEST concept for correlating weather cues across individual simulator subsystems.

With the WEST approach, environmental correlation between simulators and simulator subsystems is achievable by sharing a common numerical description of weather conditions over the gaming area (the unified weather database). Just as terrain data is shared between simulators in a common generic format (Project 2851), the WEST approach provides a similar technique for sharing weather data. The primary difference is that terrain data is static (excepting dynamic terrain), while weather data is time-varying and subject to dramatic change. This characteristic calls for a different way of thinking about how weather data should be accessed and processed

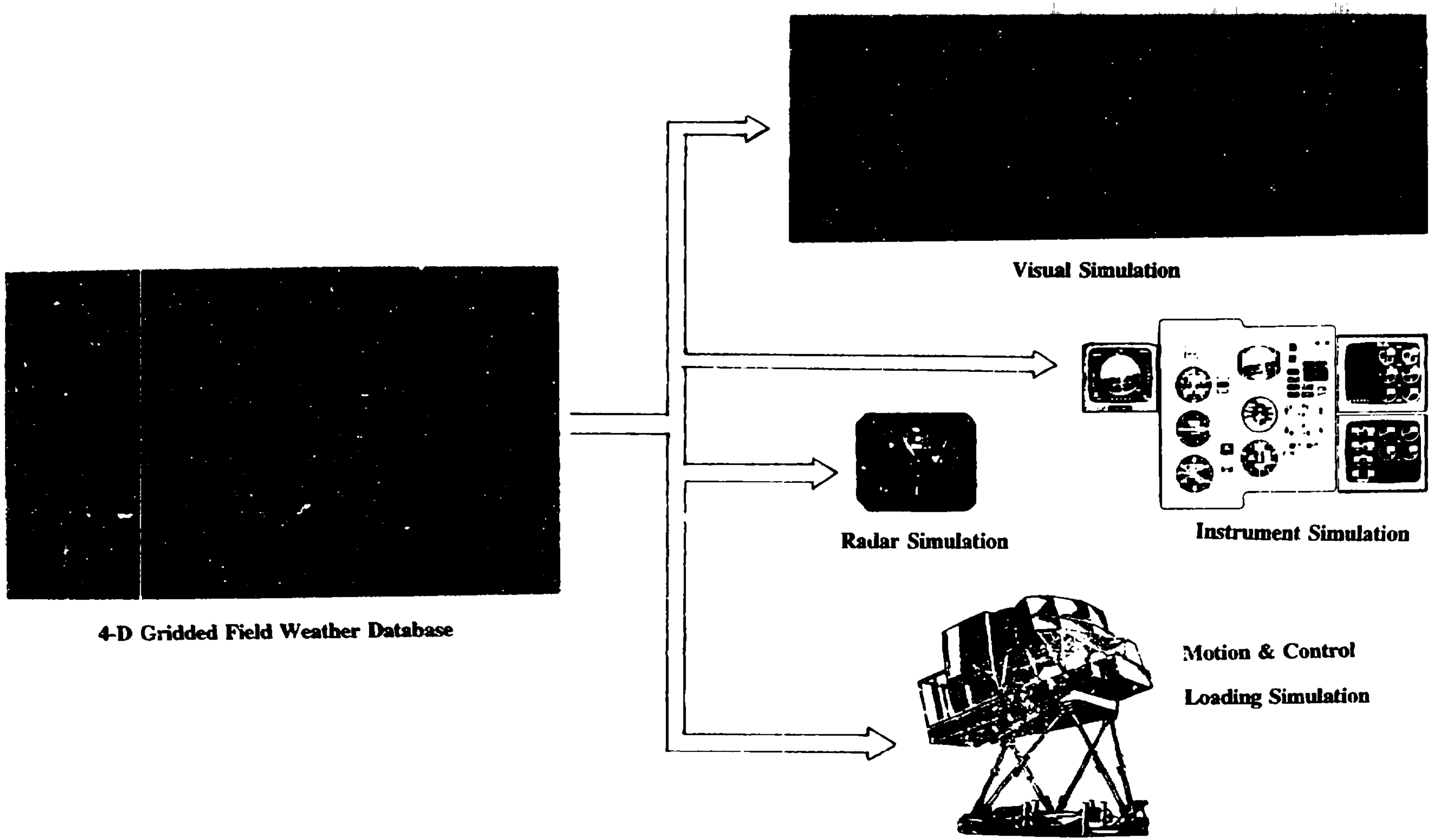


Figure 1 WEST Unified Weather Simulation

during simulator operation. With the WEST approach, each simulator subsystem is provided with only the necessary data required for immediate processing by the subsystem, the intent being to minimize the computational timing and sizing impact on subsystems for incorporating dynamic weather data. The other benefit is that with this architecture, simulators can accommodate "just in time" weather feeds for integrated live/virtual training in addition to accommodating predefined, canned weather scenarios.

Digital atmospheric data is becoming increasingly available in the form of time-varying 3-D gridded data sets. Essential simulation capabilities for generating simulator cues from gridded data sets have been outlined by the author[1,2] and include 3-D gridded-field data handling, position-independent viewing/sensing, and fly-through viewing.

Digital Data Handling Capability. In order to drive simulator subsystems directly from digital atmospheric data, the weather simulation technique must be capable of manipulating and processing 3-D gridded data very rapidly to support real-time scene update rates (typically 30Hz). The data handling technique must accommodate paging between spatial weather data subdivisions or tiles as well as continuous interpolation between adjacent time-stamped data files. This capability is essential for handling very large spatial and temporal data sets.

Position Independent Viewpoint Capability. A key requirement is that the ideal digital weather simulation technique must support totally independent viewpoint capability for sensor scan processing of the gridded weather data set. This means that the data must be sensed and formatted from any position, orientation, or relative motion with respect to the data set. This capability is essential for simulator applications where the platform is almost always maneuvering within the boundaries of the 3-D gridded field. In many circumstances for high-altitude weather viewing, the entire data set may be located within the platform's visual field of view.

Fly-Through Viewing Capability. In addition to supporting unconstrained platform motion, the optimal simulation technique for 4-D weather must also support seamless fly-through visual viewing of the data. The student should be presented with perspective-correct imagery under all viewing conditions. The visual appearance of the weather effects should automatically compensate for viewing aspect, lighting conditions, and relative motion of the eyepoint and dynamic weather effects data.

WEST DESCRIPTION

Advances in computer image-generation (CIG) processing power, coupled with the increasing availability of gridded field weather data, now make realistic weather modeling practical for use in manned simulators. During the WEST research effort, attention was focused on the visual simulation of weather. Several weather imagery techniques were evaluated on high-capacity graphics workstations to determine the best method for simulator application. An approach using continuous-level-of-detail graphics primitives was selected as the best method, and a frame-based image-generation transform function was developed for constructing and rendering scene primitives from weather database parameters in real time.

The WEST approach is specifically designed to support real-time visual simulation requirements associated with rendering 4-D weather imagery directly from digital source data. The WEST prototype[3] executes on a Silicon Graphics Crimson Reality Engine with two Raster Managers. Figure 2 illustrates the architecture and functional elements of the WEST approach. Major elements include the digital weather source database, a visual preprocessor component, a data handling component, a simulation interface component, a visual weather database, and an image-generation component.

Digital Weather Source Database

WEST is designed to process large-scale gridded data sets containing digital atmospheric data parameters that may be produced from sensor observations or from numerical models. Digital atmospheric physics models such as TASS and Doppler radar-derived atmospheric observations such as JAWS data sets are examples of the types of gridded field data sets available that provide quantitative, time-varying descriptions of aviation weather conditions.

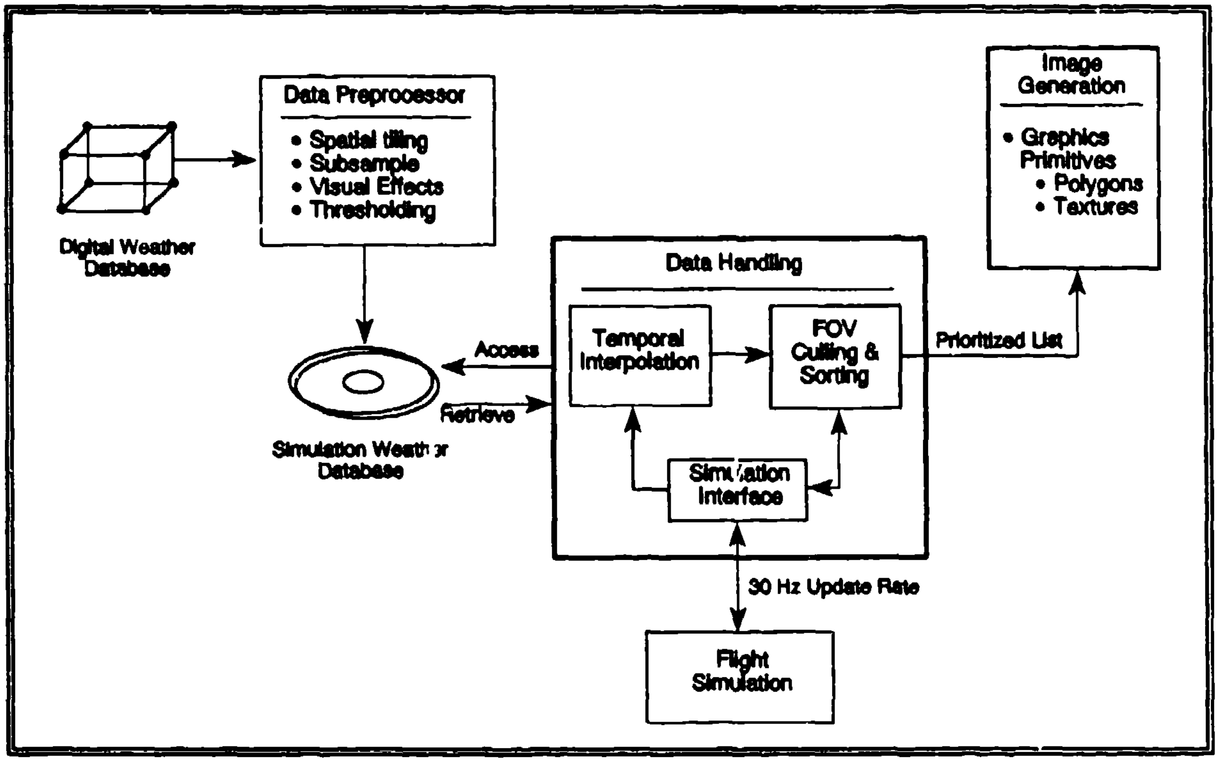


Figure 2 WEST Visual Simulation Approach

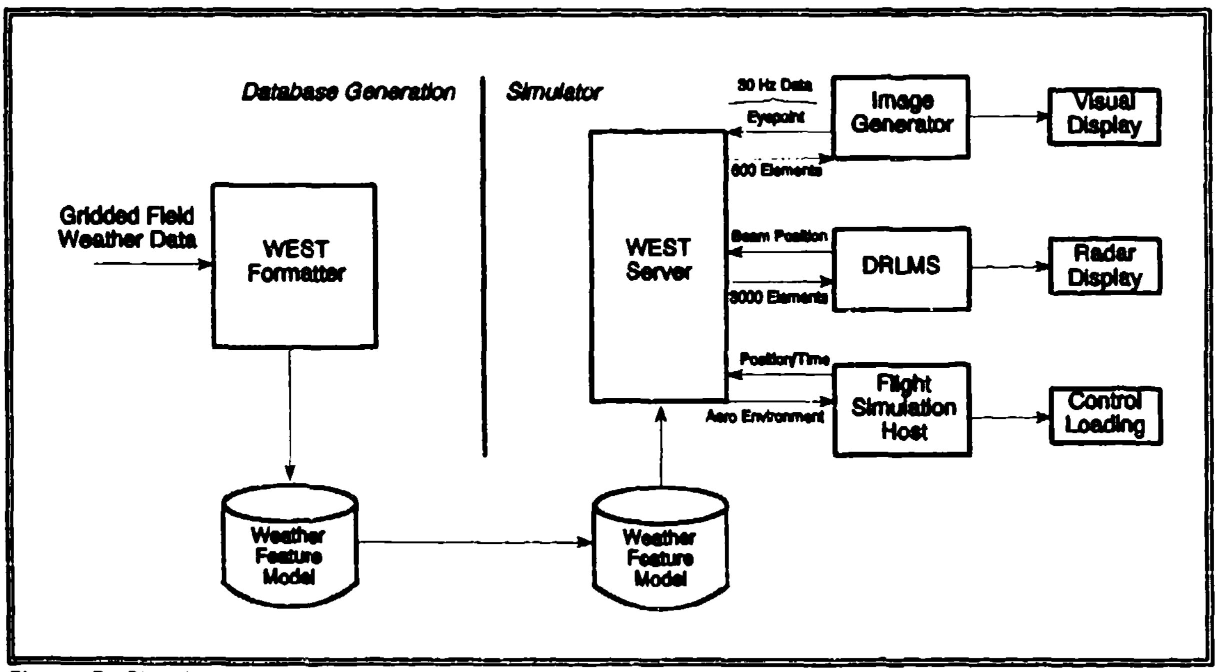


Figure 3 Simulator Integration

For the TASS model, each grid point within the model domain contains 11 atmospheric parameters describing the state of the air mass volume for that spatial location at a given point in time. These parameters include north, east, down wind speed, pressure, temperature, liquid water content, ice content, precipitation rate, snowfall, and hail fall. Grid spacing for the microscale TASS model varies between a maximum of 200 meters down to 40 meters within a typical domain size of 500 cubic miles (10mi x 10mi x 5mi).

The NCAR database contains gridded parametric data for wind speed and direction, radar reflectivity, and derived liquid water content at 200-meter grid intervals within a 16km by 16km volume of airspace. Additional types of sensor-derived data sets are typically mesoscale-oriented and contain grid spacing on the order of 500 to 1000 meters, and cover thousands of cubic miles of atmosphere. Grid domains for these data sets can extend to 2048 x 2048 x 15 elements. Weather parameters typically supported include temperature, wind speed, pressure, water content, and water type.

Visual Preprocessor Component

Preprocessing methods were implemented within WEST for reading in gridded weather database files and reformatting the data for real-time simulation. Data reformatting includes tiling the data into spatial subdivisions known as cells, and preprocessing/compressing the source data to support simulator subsystem simulations. Preprocessing operations currently implemented include spatial subsampling, data thresholding, coordinate transformation into north-east-down coordinates, liquid water content calculation, texture/primitive assignment, apparent lighting calculation, and spatial dithering. Graphic primitives are assigned to weather data elements according to a parametric data look-up table to support out-the-window scene generation. These primitives are stored in a library format and include textures as well as polygon primitives. Lighting effects are precomputed based on sun position as determined by a time-of-day, day-of-year look-up table. A graphical user interface allows the user to interactively control preprocessing functions for tailoring and evaluating the visual appearance of the resulting visual weather database.

Unified Weather Database. The unified weatner database is the run time database that has been reformatted, tiled, and compressed to support real-time simulation. The run time database is a series of time tagged files that describe volumetric weather data for a given geographic location. Each weather data element within the database contains a parameter list that describes the physical characteristics of the air mass parcel located at that element's spatial position in the atmosphere. Weather parameters maintained within the run time database on a per-element basis include wind vector, water content, spatial extent, texture assignment, color, illumination, and transparency. The unified database consists of spatial and temporal linked files that are retrieved from disk and then loaded into memory as needed during simulator operation.

Data Handling Component (Weather Generator)

The weather-generation component performs the operations necessary to process and format the volumetric weather data within a sensor's instantaneous field of view. The weather generation component is scheduled by a frame-based executive to assure periodic weather data update rates and synchronization with the flight simulation and image-generation functions. The weather-generation component controls the management and distribution of weather data to simulator subsystems. For visual systems, the weather generator controls the building of the weather display list that is converted to immediate mode display data by the image generator hardware. Weather data handling operations include data retrieval and temporal interpolation, field-of-view (FOV) culling, sorting, and prioritizing of in-FOV weather data, and formatting/distribution of active weather elements to the image-generation component. In a weapon system trainer application, this component would also format and distribute weather data within a sensor's scanning field of regard or instantaneous field of view.

Image-Generation Component

The image-generation component processes the visible weather element display list that was produced by the weather generator and transforms these weather elements into a hierarchical list of weather "objects" consisting of polygon vertices

and graphic library-specific parameters for immediate mode rendering on the image generator's graphics pipeline. The weather objects represent the visual depictions of air mass parcels containing liquid water that are assembled and blended to provide a composite weather scene. Targets and special effects (e.g., weapon impacts, smoke plumes) are inserted within the integrated and prioritized weather/terrain display list at the appropriate time to provide weather occulting and obscuration effects on visible objects.

Flight Simulation Interface Component

The flight simulation interface component receives instantaneous aerodynamic environment parameters as a function of viewing platform (aircraft) position. Weather data elements surrounding the viewpoint position are interpolated tri-linearly to determine the instantaneous atmospheric environment including the net aerodynamic effects acting on the viewing platform. These effects include north, east, down wind speed components, time rate of change of these components, and ambient temperature and pressure at the aircraft. The simulation interface component receives the viewpoint and viewing orientation from the flight simulation host computer and provides this data synchronously to the image-generation component. This capability allows WEST to be applied within flight simulators to assure direct correlation between visual weather imagery and aircraft dynamic modeling functions.

WEST APPLICATIONS

Weapon System Trainers

The WEST method for processing digital atmospheric data in real time is ideally suited for weapon system training applications. With the WEST approach, a weather generator may be integrated within the simulator configuration to process and distribute dynamic digital weather data to each atmosphere-sensing simulator subsystem, as shown in Figure 3. The number of active weather elements that are provided by the server to each subsystem is a function of the subsystem's field of regard and resolution. For visual systems, the field of regard is defined by visibility range and field of view. For DRLMS systems, the field of regard and resolution is defined by the antenna beam width, pulse repetition frequency, azimuth scan limits, and radar range scale. Weather elements from a unified digital database are distributed to each subsystem (visual, radar, flight) as a function of subsystem viewing parameters (position, orientation, field of regard). Each subsystem then performs the required function for transforming digital weather elements to display cues. The visual system transforms the weather data elements into scene primitives (as demonstrated by WEST), and the radar simulation subsystem incorporates the weather data elements into range bin processing for calculating attenuation and backscatter radar effects.

Figures 4 and 5 illustrate the visual realism available with the WEST technique. Figure 4 is an example of synthetic optical imagery generated directly from NCAR gridded-field weather data by WEST. Figure 5 shows the individual weather elements (rendered as vectors) that serve as the basis for Figure 4.

Mission Planning and Mission Rehearsal

Mission planning and mission rehearsal training could become more effective and transferable to operations through the application of the WEST approach for simulating weather conditions that are expected or forecast for a given mission. Weather environment effects on weapon system employment and combat tactics procedures could be simulated and evaluated by integrating real-world weather conditions into the synthetic mission environment provided for mission planning systems and tactical preview systems. Given the gridded-field data handling capability of the WEST approach, it is conceivable that satellite-derived weather conditions could be formatted to produce a geospecific, synthetic weather environment for mission rehearsal training. Incorporating satellite weather data for simulator training is an area of continuing research and development.

SUMMARY

This paper has presented an overview of the WEST weather simulation approach. This approach is capable of integrating real-world weather into manned simulators in such a way that display cues may be automatically correlated across simulator subsystems. This capability is achievable due to the unified format of the weather database, and simulator architecture modifications that accommodate dynamic data handling with minimum

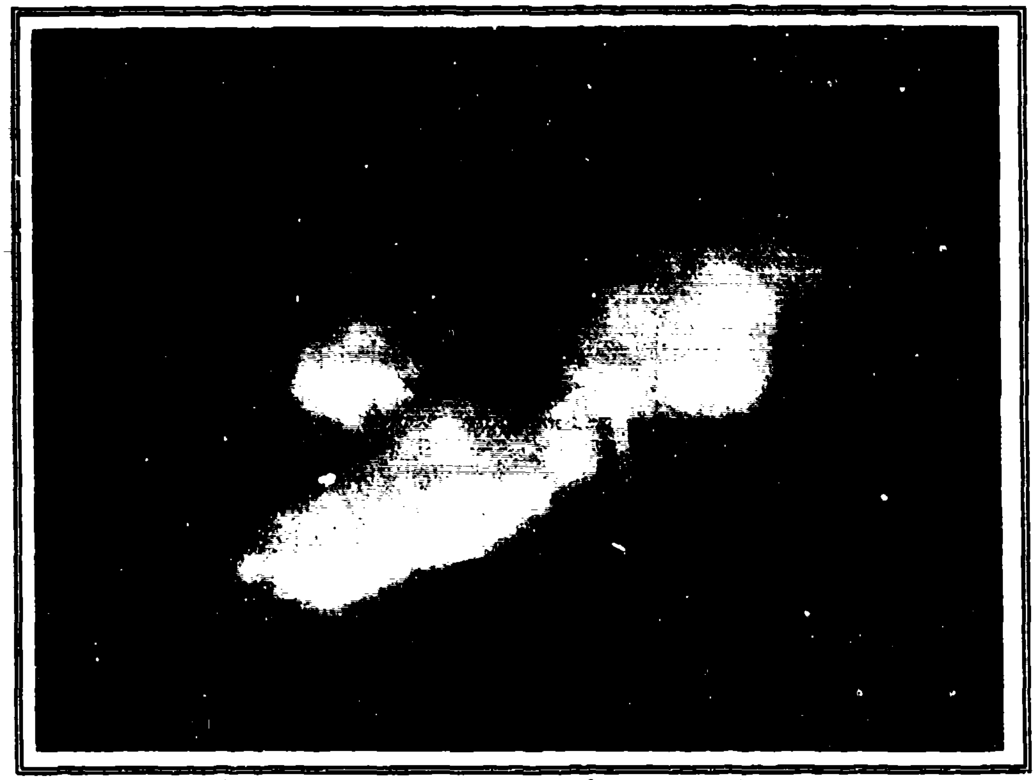


Figure 4 WEST Computer-generated Weather Scene From Gridded-Field Data

Figure 5 WEST Active Weather Elements Within Gridded Field

impact to simulator subsystems. The approach as demonstrated is extensible to accommodate the special needs of simulator subsystems, including DRLMS, FLIR, and weapon system simulations.

Although the WEST approach was developed in response to a need to improve the way in which atmospheric effects are modeled for flight training, the simulation techniques implemented within WEST are directed at solving weather-related simulation problems pertaining to sensor simulation modeling, geospecific mission rehearsal applications, and multi-force distributed network training applications.

REFERENCES

1. B. C. Montag, "Adverse Weather Simulation Concepts for Safety of Flight Training," Paper No. AIAA-90-3132-CP, AIAA Flight Simulation Technologies Conference, 1990.

2. B. C. Montag, "Visual Simulation of Digital Atmospheric Data," SPIE Aerospace Science and Sensing Symposium, April 16, 1993.

3. B. C. Montag, "Adverse Weather Visualization Modeling," Southwest Research Institute IR&D Project 05-9665 Final Report, October 1992.

HIERARCHICAL TRAINING FOR ARMY AVIATION

William A. Yarlett
Reflectone, Inc
Tampa, FL
and
Sandra M. Szabo, Ph.D.
Sikorsky Aircraft
Stratford, CT

ABSTRACT

In 1993, the Army Aviation community conducted a review of simulation and training requirements for deployable simulators and devices that support individual/crew sustainment training, collective training, and combined arms training. Following the review, the Army's Mobile Aircrew Sustainment Trainer (MAST), Future Aircrew Sustainment Trainer (FAST), and Aviation Combined Arms Tactical Trainer (AVCATT) programs all underwent scrutiny to determine if they could meet the training requirements within the constraints of today's austere budgets. This paper presents a training concept that consolidates Army Aviation simulation and training requirements under one program offering a single hierarchy of individual/crew and collective training devices. A basic tenet of the paper is that a single program could provide better training at a lower cost than several independent programs. The key to affordability resides in:

- Using state-of-the-art technology to reduce the recurring cost associated with training device hardware development.
- Tailoring training device fidelity to meet the "margin" of acceptable training for each level of training.
- Reducing non-recurring software costs by flowing software from full-fidelity training devices down to lower-fidelity devices.

ABOUT THE AUTHORS

Bill Yarlett is the Business Development Director for Rotary Wing and Mobile Trainers at Reflectone in Tampa, Florida. Before moving to Reflectone, Bill was the New Business Development Manager for Army/Navy Programs and Program Manager, Comanche Training Device System at CAE-Link Corporation in Binghamton, New York. A retired U.S. Army aviator, Bill's military career included numerous Army aviation assignments as combat pilot, instructor pilot, developmental test pilot, and operational test pilot. His last Army assignment included positions as Assistant Product Manager (APM) for Visual Systems, APM for the AH-64 CMS Program, and APM for the AH-1 FWS Program at the Office of the Project Manager, Training Devices (PM TRADE).

Dr. Sandra Szabo is an experimental psychologist with more than ten years of experience in human factors engineering and training system development for Army aviation systems. Her primary areas of expertise include mission/task analysis, operator workload prediction, and training and weapons system design. As a collocated contractor representative, she has provided technical advisory support to system program managers and developmental engineers at the U.S. Army Aviation and Troop Command in St. Louis, Missouri, and to training and combat developers at the U.S. Army Aviation Center at Fort Rucker, Alabama. Her last assignment was Manager of Training Operations for Army programs at CAE-Link Corporation in Binghamton, New York. Today, she is Technical Lead, Human Factors Engineering, for Comanche Crew Systems Design at Sikorsky Aircraft in Stratford Connecticut.

HIERARCHICAL TRAINING FOR ARMY AVIATION

William A. Yarlett
Reflectone, Inc
Tampa, FL
and
Sandra M. Szabo, Ph.D.
Sikorsky Aircraft
Stratford, CT

INTRODUCTION

Since the 1970's, individual/crew training devices have provided a means for Army aviators to acquire and sustain skills in aircraft and systems operations. During the past two decades, simulation popularity among aviators has grown substantially, due in large part to advances in technology that offer more powerful visual systems, faster computers, realistic aural cues, and improved visual displays.

During the 1970's, state-of-the-art technology provided just enough device performance to meet the margin for satisfactory training. In contrast, today's medium fidelity technology provides more training performance at a lower cost than yesterday's high-fidelity technology. Army planners and industry training developers can take advantage of this technology to meet aviation training and simulation requirements within present-day budget constraints.

Currently, several independent Army programs are attempting to fulfill the requirements for affordable and deployable devices to support individual/crew sustainment, collective, and combined arms training. In today's funding environment, a strong possibility exists that one or more of these programs may be postponed or cancelled. A solution is needed to avoid the potential degradation to training that budget cuts might cause. This paper proposes a potential solution that consolidates all Army aviation training requirements under a single program that would cost less than multiple independent programs. Specifically, the paper proposes a hierarchy of training devices that meets both individual/crew and collective training requirements for Army aviators. While training programs for new systems such as the OH-58D, Longbow, and RAH-66 would benefit most from this strategy, training devices for fielded systems could also benefit from the proposed training concept. In both cases, the potential exists to fulfill all training needs at a more affordable cost.

The training hierarchy proposed herein is based upon the authors' experience in training, training systems analysis, and training device development and has not undergone a Systems Approach to Training (SAT) process. Consequently, we propose that, prior to the implementation of any ideas contained in the paper, a full SAT analysis be conducted to confirm the value of our proposed training solution.

BACKGROUND

The Training Hierarchy

Current Army aviation training uses a building block approach in which individual/crew skill training at the school house precedes sustainment and collective task training in the aviation unit. Individual/crew training focuses on the acquisition and sustainment of skills required to fly the aircraft and operate its systems. The suite of devices used to conduct this training provides a hierarchy of device fidelity that parallels the complexity of the individual/crew training requirements. These devices typically range along a continuum from cockpit procedures and part-task trainers to high-fidelity simulators. Once in the field, aviators use the aircraft and any available training devices to sustain their individual/crew skills.

Traditionally, collective training has been taught in the aircraft after the aviator arrives at the unit. During collective training, the aviator learns to operate the aircraft, first as a member of a team (2 to 3 aircraft), then as a member of a platoon or troop, and finally as part of a battalion or squadron.

Today's Hierarchy of Training Devices

Figure 1 illustrates the current hierarchy of devices that support Army aviator training. At the lowest level of the hierarchy Cockpit Procedures Trainers (CPTs) support training in cockpit switchology, engine start, and subsystems operations. At the second level, Part Task Trainers (PTTs) provide the means to train complex, time-consuming tasks. An example of a PTT is the Army's Target Acquisition and Designation System (TADS) System Task Trainer (TSTT) used by Apache pilots to learn sensor and weapons skills. At the third level in the hierarchy, is the Operational Flight Trainer (OFT) or Flight Simulator (FS), which supports flight training and total system operation. These devices typically have sophisticated flight models, high-fidelity visual systems,

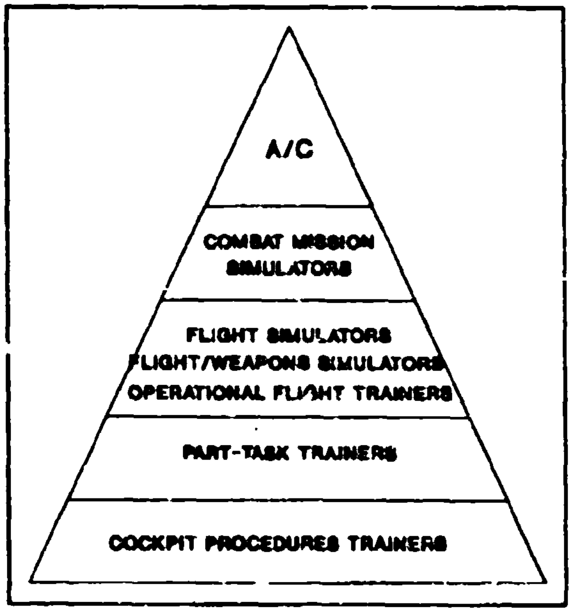


Figure 1 - Today's Individual/Crew Training Device Hierarchy

and six degree-of-freedom (6-DOF) motion systems. An OFT or FS may also support weapons training, in which case it is called a Flight and Weapons Simulator (FWS). Combat Mission Simulators (CMSs) comprise the fourth level in the training hierarchy. A CMS is an OFT, FS or FWS that includes simulation of interactive threats to enable the crew to operate their simulated aircraft in a threat environment.

The operational aircraft used to be considered the ultimate training device in the training device hierarchy and the only means to train collectively. However, due to the lack of range space, lack of an interactive threat force, safety considerations, budget reductions and cutbacks, aircraft are no longer used to sustain aviator combat proficiency on a regular basis. Until the AVCATT system is fielded, the Army does not have an aviation collective training device.

TOMORROW'S HIERARCHY OF TRAINING DEVICES

This paper proposes an alternative suite of training devices that supports the training of both individual/crew and collective skills. The devices use state-of-the-art technology to provide a level of training fidelity specifically tailored to the margin of training appropriate to each level of individual/crew and collective training. A basic premise of the paper is that a single hierarchy of devices could significantly reduce training system cost. The primary cost savings accrue from flowing software down from the highest fidelity device to lower fidelity devices and by substituting or eliminating high-fidelity hardware in the lower fidelity devices.

Figure 2 illustrates the proposed training device hierarchy. The Simulator (Enhanced) trainer, called the SIM (E) is the highest fidelity device in the hierarchy. Individual/crew training devices representing decreasing levels of fidelity extend to the left of the SIM (E) device, while collective training devices representing decreasing levels of fidelity extend to the right of the SIM (E) device.

Individual/Crew Training Devices

The proposed individual/crew training device hierarchy includes four training devices

- Cockpit Procedures Trainer (CPT)
- Part-Task Trainer (PTT)
- Simulator Enhanced, SIM (E)
- Simulator Plus, SIM (+)

The CPT and PTT devices possess the same general level of fidelity and support the same training functions as the CPTs and PTTs included in today's hierarchy. The SIM (E) device is comparable to traditional full-fidelity simulators and is characterized by high-fidelity flight models, high-resolution visual systems, and 6-DOF motion systems. However, in the proposed hierarchy, the SIM (E) trainers are used <u>only</u> to conduct initial acquisition of individual/crew skills at the school house. That is, unlike the simulators in today's hierarchy, the SIM (E) devices are not used to support individual/crew sustainment training in the units. Instead, the proposed hierarchy includes a

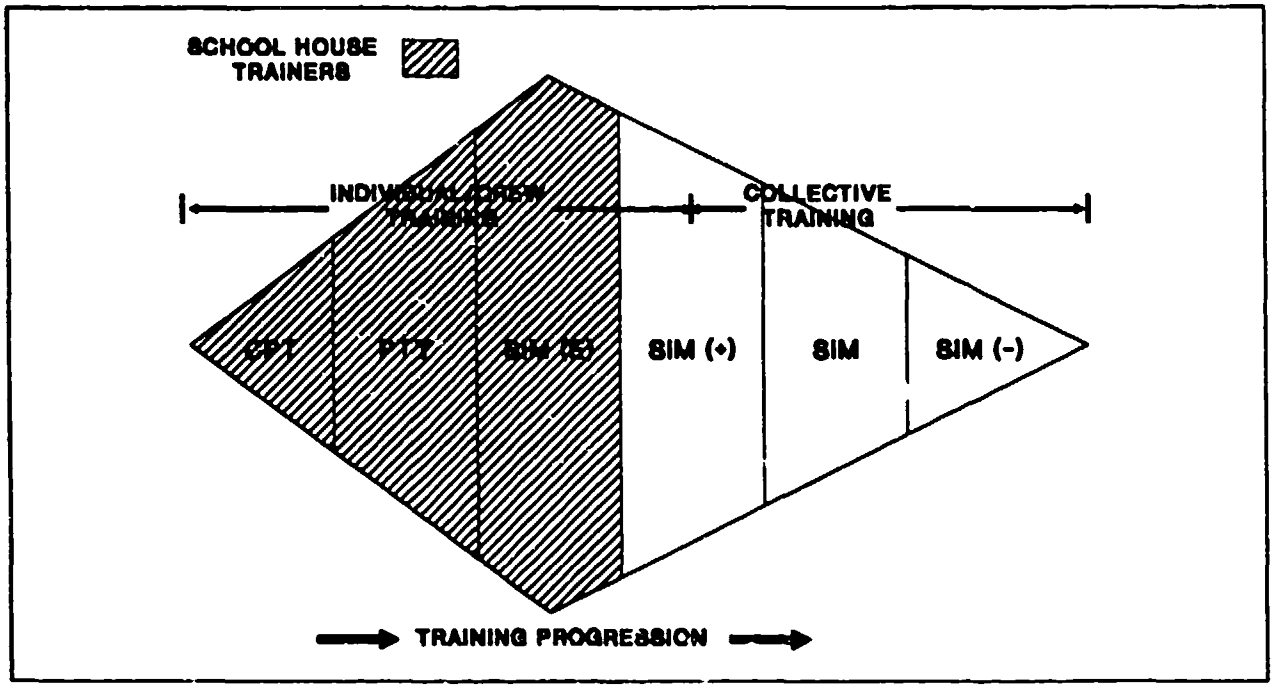


Figure 2 - Proposed Future Army Aviation Device Hierarchy

variant of the SIM (E), called the SIM (+) trainer, to conduct sustainment training.

The SIM (+) is best characterized as a medium fidelity training device whose principal differences from the SIM (E) include a lower fidelity visual system and seat-shaker motion cuing. Because of the reduced hardware associated with the lower fidelity visual and motion systems, the SIM (+) is a self-contained training device that can be housed in a government building or a mobile facility. This feature contrasts dramatically with the "brick and mortar" facility requirements associated with traditional simulators. Furthermore, it allows the SIM (+) to be deployed with the troops or to move from installation to installation, supporting training in units with low aviator populations and providing training to aviators worldwide.

Collective Training Devices

The proposed suite of collective training devices includes three variants of the SIM (E) trainer. The three variants, each representing decreasing fidelity, are designated SIM (+), SIM, and SIM (-). The SIM (+) is a "double-duty" training device and supports both individual/crew sustainment training and collective training. In the collective training hierarchy, the SIM (+) is used to support training at the lowest echelon level -- the team/platoon level. Thus, the SIM (+) provides the transition from high-fidelity individual/crew training to low echelon collective training.

The core simulator in the proposed collective training device hierarchy is the SIM. The SIM is a variant of the SIM (+) trainer with non-essential, non-mission equipment simulated as facsimile panels in the cockpit. After mastering collective team operations in the SIM (+) device, the aviator transitions to the SIM trainer. The primary objective of the SIM trainer is to support company/battalion level collective training.

The SIM training device is supported by a set of four auxiliary work stations, called the SIM (-) team set. The networked SIM (-) work stations enable additional pilots to enter the collective training environment, thereby increasing the number of players in the loop for any training scenario. The SIM (-) is designed to provide year-round unit collective training at the aviation battalion/squadron level.

Figure 3 presents a comparison of the hierarchy of training devices currently used to conduct Army aviation training and the hierarchy of devices proposed herein.

Why a Training Hierarchy

During training, an aviator progresses sequentially through two hierarchies of training. The first hierarchy is individual/crew training where the aviator learns to fly the aircraft and operate the systems. As the aviator progresses through the levels of individual/crew training, the complexity and cost of the individual/crew training devices increase in <u>direct proportion</u> to the level of training performed.

Having acquired the necessary individual/ crew skills, the aviator then progresses to the collective hierarchy of training. Collective training provides a means for the aviators to learn to operate their aircraft on the battlefield as a member of the combined arms team. Collective training should occur in stages of progressively higher echelons beginning at the team level and progressing to the company, battalion, squadron, brigade, and higher levels. The required level of fidelity for collective training devices is <u>inversely proportional</u> to the echelon level of collective training.

Using the Hierarchy to Train

Having completed initial acquisition training in individual/crew skills at the school house, an aviator arriving in the unit is already qualified to fly the aircraft and operate its systems. The SIM (+) version of the high-fidelity SIM (E) device provides sufficient fidelity to sustain the individual/crew skills of the unit aviators. As a derivative of the SIM (E) trainer, the SIM (+) includes most of the software and training potential of the SIM (E), but at a significantly lower cost.

Upon entry into the collective training hierarchy, the aviator begins training at the lowest echelon, the team level, and progress to higher echelons. In the proposed training device hierarchy, team training is conducted in the highest fidelity collective training device, the SIM (+). The SIM (+) device imposes total aircraft and subsystem operation responsibility on the aviator while simultaneously introducing the skills required to coordinate the mission as a member of an aviation team. An advantage of the SIM (+) device is that it minimizes the negative transfer of training in basic flight skills the aviator might experience during the transition from individual/crew operations to collective team operations. That is, the SIM (+) bridges the gap between the high fidelity flight and systems models in the individual/crew training devices and the reduced fidelity models that characterize the higher echelon collective training devices. Gradual transition into the lower fidelity devices enables the aviators to learn to cope with the additional demands of collective mission operations without degrading their individual/crew skills. Although the SIM (+) objective is to support team level collective training, it can be networked to other compatible trainers to support company/ battalion level training. Higher echelon training with the SIM (+) is conducted on a limited basis and only to reinforce the complexity of collective training in a full fidelity aircraft.

		SCHOOL HOUSE				UNIT
INDIVIDUAL/ CREW TRAINING	TODAY	CPT	PTT	OFT/FS FWS	CMS	OFT/FS/ FWS/CMS
	TOMORROW	CPT	PTT	SIM (E)		SIM (+)

		TEAM TRAINING	COMPANY/ BATTALION TRAINING	SUPPLEMENTAL TRAINERS/UNIT TRAINERS
COLLECTIVE TRAINING	TOMORROW	SIM (+)	SIM	SIM (-)

Figure 3 - A Comparison of Hierarchies, Today and Tomorrow

As the level of collective training increases to the company and battalion echelons, the training emphasis changes from aircraft and system operation task loading to higher echelon decision making and improving the aviator's ability to impart cause and effect on the battle. Having gained an appreciation for the complexity of both aircraft operation and mission responsibilities during team collective training in the SIM (+) device, the aviator can now progress to training at the next level of collective operations. The SIM trainer is the core training device for company/troop level operations. Like the SIM (+), the SIM may be networked with other training devices to provide collective training at higher echelon levels.

In the company/battalion level collective training scenarios, the number of players is limited by the number of networked SIM (+) and SIM devices. The number of networked devices, in turn, is constrained by budgets and training throughput. The SIM (-) is proposed as a low-cost device that increases the number of players who experience year-round collective training. The Sim (-) provides a set of four networked work stations supporting four pilots operating as a team. SIM (-) operators may also include staff officers observing the mission from stealth vehicles (vehicles not seen in the visual scene).

The SIM (-) work stations provide an enroute flight mode to teach maneuver skills and a battle position engagement/observation mode to teach weapons and sensor operations. With SIM (-), teams can train independent of SIM training or supplement to SIM training. The independent mode enables every aviation unit to have year-round collective training, sustaining a higher level of mission proficiency and providing greater benefits from annual collective training exercises. The SIM (-) team set would become part of the aviation battalion/squadron Table of Organization and Equipment (TOE).

The Cost Savings

The high costs of operating aircraft makes simulation a cost-effective component of Army aviation training systems. Although all levels of training can be conducted in a single high fidelity device (CMS, FS, or OFT), a hierarchy of devices enables many training requirements to be fulfilled in lower cost devices (CPTs and PTTs). Off loading training requirements to lower fidelity devices means fewer CMS or FS device hours are needed, reducing both the acquisition and life-cycle costs of the total training system.

The potential savings realized by building a hierarchy of devices are enormous. Most of the software originating in the SIM (E) device is common across the device hierarchy. From the SIM (E) starting point, hardware is deleted or alternative lower cost hardware is substituted in each lower fidelity device in accordance with the training requirements. This approach to training system development results in a total cost for all trainers that is less than the cost of building a single, high-fidelity sustainment training device and a different collective training device under separate programs.

Device Cost and Fidelity Differences

In the proposed training device hierarchy, CPT and PTT devices will support individual/crew training requirements as in the past. The fidelity and cost of CPT and PTT devices in the future hierarchy are comparable to similar devices included in the present training hierarchy. The SIM (E) device will replace the FS, OFT, FWS, and CMS in the future school house. New technology for high-fidelity visual and motion systems should accommodate a recurring cost for SIM (E) at half the cost of past CMSs.

In the future hierarchy, the SIM (+), will pull double duty as an individual/crew sustainment trainer and an introductory team collective trainer. The SIM (+) software loads will be identical to the SIM (E) software; however, SIM (+) will have less motion fidelity than the SIM (E) device and a lower cost visual system. The recurring cost of the SIM (+) device will be 40 percent of the cost of the CMS device it will replace.

The SIM device is the core collective trainer in the future training hierarchy. The SIM is characterized by minimum motion cuing, a reduced cost visual system, and facsimile panels for non-essential cockpit controls and displays. Consequently, the recurring cost of the SIM device will beone-third the cost of a SIM (+) device.

A SIM (-) device consists of a set of four work stations that provide an enroute flight mode and a battle position engagement or observation mode with auto-pilot. The SIM (-) is fully reconfigurable to support the simulation of flight, weapons and sensor systems for all aircraft types. The weapons and sensor system software for SIM (-) is flowed down from the SIM (E). The affordability of SIM (-) will enable 24 additional aviators to participate in a collective training scenario in six SIM (-) team sets for the same recurring cost as a single SIM device.

Design to Cost Goals

To achieve the cost reductions for the proposed training devices, design to cost (DTC) goals must be established for major simulator subsystems. Examples of two major subsystem cost drivers, visual image generators and motion systems, are presented below. The DTCs established for each subsystem include the cost of state-of-the-art technology available today. These cost goals describe system performance at the training margin for each level of training. Higher cost systems may exceed the margin for individual/ crew training and collective training needs.

- Visual Image Generator (IG): 6 channels, 2,000 polygons per channel; 1M pixel resolution with special effects, weapons effects and 60 Hz iteration rate. Recurring DTC: $2M, SIM (E) and SIM (+) devices.
- Visual Image Generator (IG): 4 channels, 2,000 polygons per channel; 750K pixel resolution with special effects, weapons effects and 60 Hz iteration rate. Recurring DTC: $1.2M, SIM device.
- Visual Image Generator (IG): 4 channels, 1,000 polygons per channel; 750K pixel resolution with special effects, weapons effects and 30 Hz iteration rate. Recurring DTC: $200K, SIM (-) device.
- Motion Cuing: Recurring DTC: $1.2M, SIM (E) device.
- Motion Cuing: Recurring DTC: $300K, SIM (+) device.

SUMMARY

With state-of-the-art technology, the costs of simulation and training devices are slowly coming down. To control costs, it is important to define the margin for training performance and to select the appropriate technical solution. An alternative approach is to establish both a training margin and a DTC for each required trainer. This approach would allow (1) industry to bid the bestperformance possible within the DTC budget and (2) the Army to select vendors who offer the "best value training solution. Otherwise, the Army must select the vendor who meets the margin of training performance at the lowest cost.

A hierarchy of trainers can significantly reduce the overall non-recurring costs, especially the costs of software development. Design work is also reduced if all devices are a variant of a higher order device. The cost benefits attributable to design commonality, hardware modularity, and life-cycle support savings afford an opportunity to meet current training requirements within the constraints of today's austere budgets.

Additionally, the hierarchy proposed herein ensures that all levels of training can be provided to all aviators. Collective trainers designed to be mobile can be deployed with the troops or relocated from one installation to another. Mobile training devices provide a particular advantage for National Guard and Reserve aviator training. Containerized training devices also appreciably reduce building construction costs.

A CONCEPTUAL ARCHITECTURE FOR INTEGRATING TACTICAL ENGAGEMENT SIMULATIONS (TES)

James T. O'Connell, Jr.
O'Connell and Associates, Norfolk, VA
Larry L. Mengel
LL&M Associates, Elizabethtown, KY
Dr. Thomas A. Mastaglio
***PAIDEIA*, Virginia Beach, VA**

ABSTRACT

In this paper the authors' suggest a conceptual architecture for achieving the high level of integration required to insure fidelity and tactical realism in the environment of a synthetic electronic battlefield. The architecture focuses on the concepts associated with the development of a family of knowledge-based software modules that populate the battlefield. These modules, ACTORS, AGENTS, and Filters provide the capability required to implement the full range of functions inherent in modern tactical warfare. The approach maintains strict adherence to all of the salient features of the Army's collective training strategy. Flexibility is provided to ensure implementation consistent with current doctrine Modification can accommodate doctrinal, weapons systems and other external changes. The architecture provides an innovative design that applies current and emerging technologies to satisfy the training community's vision of a capability to integrate, in an electronic environment, the full range of tactical engagement simulations.

THE AUTHORS

The authors of this paper, all graduates of the United States Military Academy and retired Army officers, have had extensive experience in the design, development and implementation of Tactical Engagement Simulations (TES). They offer a unique understanding of user requirements and the technical foundation essential to leverage emerging technology to enhance training effectiveness/combat readiness.

Jim O'Connell received his Masters Degree in Operations Research at the Naval Postgraduate School. His military career as an Infantry officer included assignments in support of the development of tactical simulations, training and training related systems. His most recent efforts have been in support of continued development of the Army's Combat Training Center (CTC) instrumentation systems and training devices.

Larry Mengel holds a Masters Degree from Lehigh in Industrial Engineering. His active duty career as an Armor officer included command, staff, and research and development assignments . He served as TRADOC Systems Manager for the Army's Combined Arms Tactical Trainer (CATT) program. He currently serves as a consultant to industry and government on advanced applications of simulation capabilities to support training needs and other functional areas.

Dr. Tom Mastaglio received a PhD in Computer Science at the University of Colorado in 1989. His military experience includes more than 20 years in both Field Artillery and training development assignments. Tom worked as an independent consultant and research analyst prior to assuming his current position as the Deputy Program Manager for Training Effectiveness on the Close Combat Tactical Trainer (CCTT).

A CONCEPTUAL ARCHITECTURE FOR INTEGRATING TACTICAL ENGAGEMENT SIMULATIONS (TES)

James T. O'Connell, Jr.
O'Connell and Associates, Norfolk, VA
Larry L. Mengel
LL&M Associates, Elizabethtown, KY
Dr. Thomas A. Mastaglio
***PAIDEIA*, Virginia Beach, VA**

INTRODUCTION

The conceptual architecture presented in this paper provides a focal point for accelerating needed development of a technology leap-ahead to integrate tactical engagement simulations (TES) into an effective and efficient system. The overall concepts discussed herein apply to current simulators and simulations deployed in support of combat arms training today and the more technically advanced training systems currently under development. The fundamental motivation for such an architecture is to encourage the timely development of a robust, effective, and efficient electronic environment to support collective training needs.

The overall approach addresses the following needs: an expanded capability to "Train As We Fight;" enhancements that support multi-echelon training ; and, integrated linkage to a sophisticated real time After Action Review (AAR) process.

We advocate satisfying these needs through the development of an overarching computer and networking architecture that exploits application of advanced technologies to enhance system effectiveness, fidelity, and tactical realism. We submit that a computational architecture based on an object-oriented system design (OOD) is essential and we provide one in this paper. The design we advocate incorporates an engagement-based adjudication of combat and provides for the seamless integration of weapons system and other simulators, simulations and TES used on instrumented ranges. *Attrition-based combat calculi cannot seamlessly support requirements to model entity level combat elements and support higher formations which are aggregates of those elements.* Representation of combat elements as knowledge-based entities is key to achieving a technology "leap ahead" in replicating ground combat for the multi-echelon simulation we envision. We clearly recognize that additional computational resources are required to implement the concept we advocate. We believe, however, that technology growth will overcome this limitation and that the performance needed to support the computational architecture described herein will be affordable within 5 years.

Train as We Fight

To simulate realistic battle conditions during training exercises, soldiers should perform wartime tasks using their normal operational equipment. This enhances believability and supports the principle: "train as you will fight." It is important for the simulation system to be robust enough to allow all personnel participating in a training event to carry out their respective wartime functions. This is particularly true for support of command and staff training. This factor should be recognized in the design and development of embedded training capabilities in the next generation of tactical equipment.

Multi-Echelon Training Support

Our conceptual architecture assumes an electronic battlefield that supports realistic collective training at all organizational levels. We assert that a Battalion Commander should be able to train his staff in a "stand-alone" configuration or be easily linkable over an appropriate computer network to participate realistically in Brigade and larger formation exercises

Design, development, test, and implementation of a realistic multi-echelon capability would not, in and of itself, accommodate all of the requirements of the Combined Arms Training Strategy (CATS). There must be a capability to link related training events, regardless of the configuration of the TES support being provided to the participants, in a coherent and highly credible fashion. This capability must be developed to support unit training focused on a common scenario and characterized by a realistic integration of: (1) players using dissimilar simulations (e.g. CSSTSS, TACSIM, BBS, JANUS); (2) units conducting maneuver training at a Combat Training Center (NTC, CMTC or JRTC); and, (3) commanders and staffs involved in BCTP exercises.

Quality AARs Enhance Training Effectiveness

We advocate comprehensive AARs based on the Battlefield Operating Systems (BOS) and functional structure of the Blueprint of the Battlefield. Developing a high quality *feedback* capability is totally consistent with our overall concept. We would suggest that the best approach here is to build on the previous developments of the AAR processes that support the CTCs.

We anticipate an AAR support capability that provides the results of engagements and events efficiently and without bias. Data will be available to commanders and controllers "on demand" explicitly describing *what* happened and supporting the analytical determination of *why* it happened. This data package will assist in preparing and conducting unit- (e.g. Brigade or Division level) and functionally-oriented AARs. Our approach supports both on-site AARs and take-home or electronically delivered products tailored to the training unit's requirements.

FIDELITY TO PRINCIPLES

The conceptual architecture described herein is offered as an approach to enhancing realism in a dynamic, realistic, synthetic electronic battlefield environment. Battles can be simulated at all echelons of command without human intervention or system initialization. Tactical realism within the simulation can be enhanced by integrating various man-in-the-loop capabilities available in currently fielded and developmental tactical engagement systems (TES).

We based our conceptual architecture on the following principles:

<u>Resolution of engagements is always at the entity level</u>. An entity, defined as a "killable" platform, may be a tank, helicopter, individual soldier, or communications node. *An entity is never an aggregation of platforms.* Entities interact on the battlefield according to behavioral rules, using performance data representative of observable real world actions. Adherence to this principle supports constructing a simulation that is both "intuitively" believable and inherently realistic.

<u>Linked networks enhance the player interaction and control</u>. A set of Local Area Networks (LAN) connect elements residing at a common physical location. Wide Area Networks (WAN) connect multiple LANs, or even other WANs when necessary. Our analysis reinforces the observation that not every entity or element on the battlefield (i.e., in the training exercise) needs to "know" everything that is going on in the entire area of operations. Information will be selectively provided to commanders and units depending on their distance from a particular action, their need for the information and the probability of their receiving it.

<u>Players access the simulation using a variety of modes</u>. Computer workstations, manned simulators, tactical weapons systems, and constructive TES systems provide the portals, or electronic gateways, through which soldiers, operating at different organizational levels, can participate simultaneously in supported training exercises. An architecture supporting *seamless* operations, in single or multiple modes simultaneously, is both feasible and practical by the end of this century. We envision entities capable of being both assembled and combined, as necessary, in order to meet tactical requirements. Efficient filtering allows thousands of participants to be linked and interact.

The Blueprint of the Battlefield guides and disciplines implementation. The Battlefield Operating Systems (BOS) enumerated in the Blueprint of the Battlefield support a paradigm for effective management of the simulation and support for AARs. The Blueprint provides a rational, hierarchical structure that ensures that all functions are considered both during operation of the simulation and when AARs are conducted.

ARCHITECTURAL COMPONENTS

The major components of the architecture described herein are knowledge-based software modules: a set of ACTORs, a set of AGENTs, and a set of Filters. These modules populate a realistic, three dimensional synthetic electronic battlefield. Collectively, the modules support both the simulation of ground combat and an AAR capability. The information required to support AARs is collected by the AGENTs.

Table 1 includes a description of the intelligent modules, their source of knowledge, and the respective role each assumes within the simulation.

MODULE	KNOWLEDGE SOURCE	ROLE OF THE MODULE
ACTOR	Computer Knowledge-Base	Interpret entity actions
Player Entities		
Manned Simulations	Human Intelligence	Execute tasks in a virtual environment
Instrumented Vehicles	Human Intelligence	Execute tasks in an actual environment
Intelligent Autonomous Entity (IAE)	Computer Knowledge-Base	Emulate task execution in a virtual environment
Gross Intelligent Autonomous Entity (GIAE)	Computer Knowledge-Base	Emulate systems in a virtual environment
AGENT	Computer Knowledge-Base	Assess entity actions
Filter	Computer Knowledge-Base	Manage Inter-network data flow

TABLE 1. Intelligent Components of System Level Architecture

The entities are intelligent; some have computer knowledge-bases, others contain a human knowledge source. Our proposed implementation consists of the software, databases, and computational tools needed to archive and document the outcomes of the engagement-based combat actions; ensure exercise control; and support the evaluation process. It recognizes the constant requirement to provide "near real time" analysis and feedback to exercise controllers, unit commanders and participants.

ACTORS

Our top level approach and architectural design, features a set of computational objects which we have labeled ACTORS. ACTORS enable the simulation to use entity level actions generated by: intelligent autonomous entities (IAEs and GIAEs); simulators (e.g. SIMNET,); and TES devices (e.g. MILES II, SAWE-RF, AGES II) to interface with one another and have an impact on the outcome of simulated engagements.

ACTORs are knowledge-based. Their purpose is to represent and interpret the behavioral actions of the three types of entities. They have no direct role in the battle and exist only to support the simulation, exchange information, and enhance credibility. They receive behavioral actions as input from the entities for which they are a cohort and have the ability to *decide* how to present those activities on the network to other ACTORs in the system. A suitable architecture for the ACTORS will allow them to be *tuned* to exercise conditions and modify their behavior over time through machine learning methods.

ACTORs provide the interface between the entities and the rest of the virtual battlefield. They have the ability to modify or degrade the actions and activities of the combat entities they represent. They also provide the capability to model the fog of war based on the "human factors" aspects of a unit's posture and capabilities.

The internal architecture of an ACTOR consists of intelligent software components, required knowledge-bases, algorithms, data sets, process control routines, etc., needed to perform this complex function. Their specific implementation will be developed in more detail during concept development. There are some crucial aspects of their design which can already be identified.

A software component which we call a "tactical equalizer" (analogous to the graphic equalizer in a stereo system) permits tuning the output of the entities prior to their behavior being input to the actual simulation. This tuning process depends on a model of the capabilities and status of the unit with which an ACTOR is affiliated. This guides an ACTOR in modification of the behavior of the subordinate entities of that unit and adds significant realism.

Sophisticated knowledge-bases assist the ACTOR in performing the tactical equalizing role. These knowledge bases are designed to provide an understanding of tactical methods, individual and crew performance factors, and even the effects of fatigue. Our approach uses a set of heuristic knowledge similar to an effects table.

Each ACTOR module must have the capability to make unrestricted distribution of information about the exercise to the controller and receive guidance regarding modifications to exercise conditions. Requirements for voice recognition and voice generation software could be algorithmically accommodated at the ACTOR interface.

Player Entities

ACTORs interpret the actions of three types of entities: manned simulators, instrumented weapons systems, and intelligent autonomous entities.

Manned simulator-entities are man-in-the-loop emulators of combat systems, such as those found in the Close Combat Tactical Trainer (CCTT) system. These are naturally, as opposed to artificially, "intelligent" because the behavior of a simulator entity results from the human reasoning process. The intellects of the crew members are the sources of knowledge and performance. Similar expertise must be embedded in the computer knowledge bases of the Intelligent Autonomous Entities (IAE).

Due to the "clean environment" in which simulators operate, as compared to the "dirty battlefield" simulated, the system must attenuate the actions of simulator entities. This is accomplished by the ACTOR assigned to "cohort" an entity which is responsible for passing information onto the battle network . A simulator exchanges information with an ACTOR using DIS standard Protocol Data Units (PDU). ACTORs not only receive but also comprehend PDUs and are able to, when necessary, modify a message before "broadcasting" it over the appropriate battle network.

Instrumented weapons systems entities are used to integrate subsistent TES. Data will be processed by an ACTOR and forwarded onto the main simulation network where the overall battle is portrayed. Whether this data can be extracted directly from existing range instrumentation systems or adding additional hardware is an implementation design decision beyond the scope of this paper. The ACTOR concept is critical for integrating the engagement-based data from instrumented weapons systems entities into the overall simulation.

The *Autonomous entities* are knowledge-based software modules which can be used as one-to-one replacements for simulator entities to populate the rest of the battlefield. We suggest the following two types: (1) intelligent autonomous entities (IAE) and (2) gross intelligent autonomous entities (GIAE). They could be based on the Semi-automated Forces (SAFOR) approach pioneered in SIMNET or an alternative approach such as the one which will be implemented in CCTT. Autonomous entities encapsulate the knowledge required to emulate the appropriate physical and

tactical behavior at a system platform level. An IAE emulating an M1 tank, for example, will perform as if a fully trained crew were operating their weapons system. Attenuation by the ACTOR of the actions of an IAE emulating a tank is a major feature of our concept.

The internal software structure of an IAE will encapsulate the knowledge required to insure it acts with believable behavior. Its actions will replicate an actual manned system with sufficient realism to interact in virtual space with a participant in a manned simulator. This knowledge will include proper tactical and crew procedures and human performance heuristics.

GIAEs are a simplification of IAEs. They are used to populate the battlefield in areas of the simulation that do not include man-in-the-loop entities. It is likely that they will contain the same knowledge as IAEs but apply it in a less complex manner. Observation of the portions of the battlefield containing only GIAEs will reveal a less sophisticated image and more stylized behavior. This will not degrade the simulation's ability to produce realistic combat outcomes or to support consistent training at multiple echelons of command.

The long term goal is to replace all GIAEs with IAEs. This will occur when the requisite growth in computational power is cost effective enough to allow the entire simulated battlefield to be completely populated with IAEs. The ACTORs are the most technically advanced feature of our architecture. Further research, development, and testing are required to implement the ACTOR concept.

AGENTS

The second essential component of our architecture is a structured set of knowledge-based AGENTs. The structure is based on the Blueprint of the Battlefield. The Blueprint provides a hierarchical definition of combat functions at the tactical, operational, and strategic levels. This provides an appropriate baseline for analyzing and integrating combat, combat support, and combat service support functions.

At the lowest functional level, an AGENT encapsulates the knowledge to observe simulated combat and infer whether or not a player has adequately performed the function for which the AGENT is responsible. Making this type of assessment requires synthesis of a significant amount of information. Applications of Artificial Intelligence (AI) techniques such as advanced pattern recognition, concept formation, and case-based reasoning will require significant computational processing capability.

Functional level assessments will be passed to the next higher AGENT in the hierarchy. Individual AGENTs fuse and integrate input from subordinate AGENTs within their respective "chain of command." This results in an assessment of a unit's performance at that level. This process percolates up through the hierarchy of AGENTs — one of which is analogous to each Blueprint element — until an overall BOS-based performance assessment can be made.

FILTERS

Filters are the mechanism that allows the networking of technology-supported training without attempting to expose all elements of the simulation to all of the data generated at the entity level. Filters interface lower level networks on which smaller, higher resolution engagement-based battles are played with higher level networks which represent the command and control of these battles.

Filters provide the ability to manage and control data "traffic" both on and between networks. Filters encapsulate knowledge of what information to pass between the networks. On a case-by-case basis, they modify, simplify, or elaborate on messages prior to passing the information to higher, lower, or adjacent networks.

Filters are a necessary feature for allowing entity level combat adjudication under the constraints of near term affordable network and computational resources. At some future point, when computational and network capacities have larger limits, it is foreseeable that the conceptual architecture can collapse into one large network without any filtering modules. This is a long term goal.

AAR Support

We envision an AAR process that provides full support for the feedback and evaluation metrics essential to meet the Army's collective training needs. Computational support would include a suite of integrated computer hardware and state-of-the-art software tools configured to meet the specific needs of the controllers, analysts and support personnel who use them to plan and present the AARs.

Conceptually, the AAR support system generates products based on the assessments made by the AGENTs. Graphical displays and analytical tools will generate summary statistical data. Presentation tools support developing the high quality, multimedia products required.

The AAR process supports the analysis and synthesis of information before, during, and after each appropriate training exercise. We anticipate AARs taking place both during and upon completion of the training event. Replay of segments of battles and other combat events will be available in a visual display. Recreation of the battle is achieved using the stream of Protocol Data Units (PDU) that were generated during the exercise and stored in a relational database component.

SYSTEM LEVEL OVERVIEW

The overall functional relationships among the elements and components of the simulation and the principal information flows and feedback loops are shown below in Figure 1.

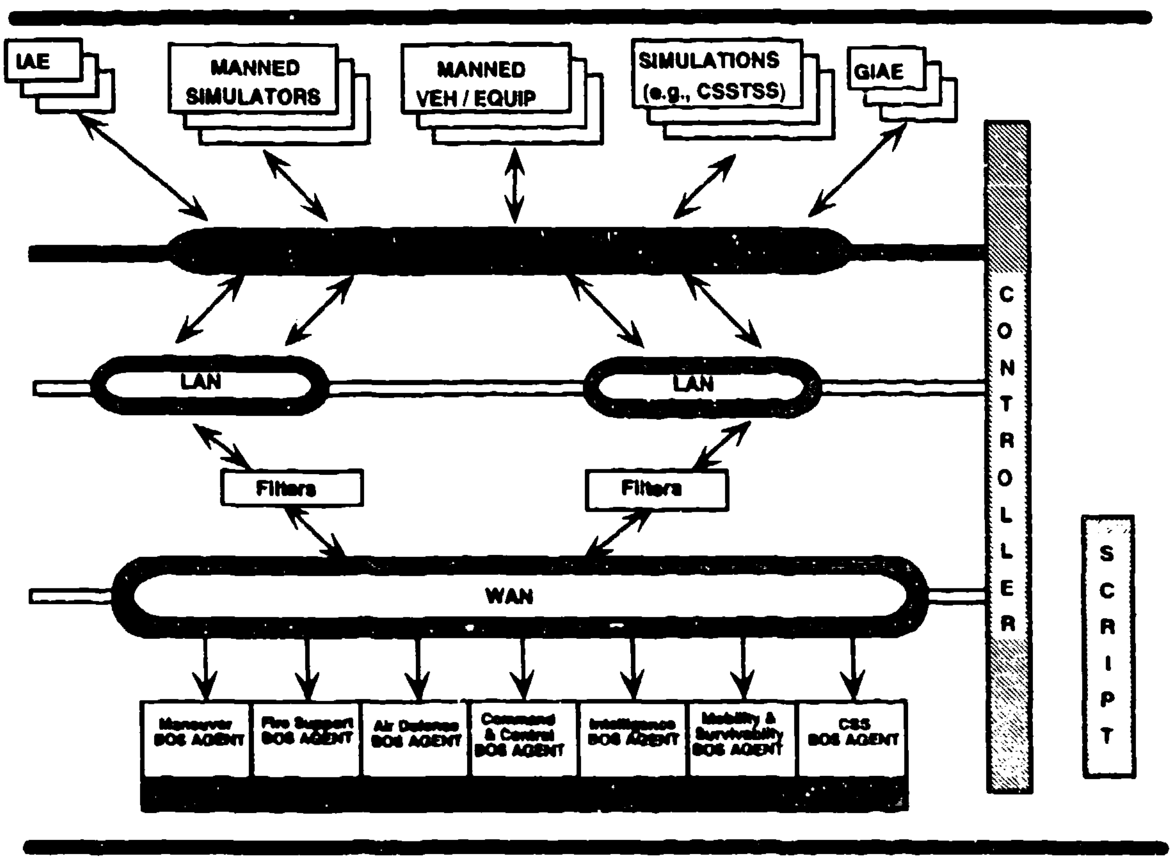


Figure 1. An Overview of the System

Manned simulators such as the Army's Close Combat Tactical Trainer (CCTT), the Air Force's F-16 trainer, or the Navy's Import Trainer would interface using DIS standard protocols.

Manned vehicles and equipment are tactical equipment with embedded or appended electronic interfaces to the simulation using DIS protocols.

Instrumented weapons systems, such as those at the National Training Center (NTC) facilitates the integration of units conducting force-on-force training. A brigade training at NTC could be "connected" with the constructive synthetic battlefield and actions of individual vehicles and units would be replicated in the simulation in a realistic and credible manner.

Simulations include existing models that support the functionality of various BOS-focused aspects of the battlefield, such as intelligence, maneuver, or combat service support. They add a level of detail and robustness in their functional areas that can be used to "stimulate" the exercise and concurrently enhance training realism.

The ACTORs are knowledge-based software components that populate the *ACTORS interface*. They monitor, analyze, and tune behaviors of player entities.

Local Area Networks (LAN) connect ACTORs within a particular geographic area and through them conceptually link entity level players. The number of ACTORs on a LAN depends on the exercise design, the network traffic load, and the network capacity. Networks also link the controller and AAR support elements to the simulation.

Filters are knowledge-based software components that determine what information must be transmitted to, or received from, another network. Filters are transparent to training exercise participants and exist only to maximize the efficiency of the networked architecture. If network capacity was large enough, or if an exercise was small enough, Filters would not be needed.

Wide Area Networks (WAN) carry information that must be shared among LANs. Although the schematic shows only one WAN, several could be linked in a hierarchical structure to support larger exercises.

Exercise control is accomplished through human-computer interfaces. The controller element monitors the exercise to ensure that it runs "fairly," interjects circumstances required by the script, and modifies exercise parameters to ensure training objectives are satisfied. Controller workstations provide a "god's eye" view of the exercise and access to ACTORs which can carry out controller instructions. The control element has the capability to communicate with all other elements in the simulation.

The AAR function, supported by AGENTs that monitor network traffic, collect battle information and analyze that information, is designed with a dedicated AGENT for each element in the BOS hierarchy. Files are built concurrently with the progress of the exercise and are able to provide up-to-date summary information on demand throughout an exercise. At the end of an exercise, this systemic technical support helps a unit commander organize and present AARs.

THE SYNTHETIC THEATER OF WAR

The synthetic environment shown in Figure 2 summarizes how our architecture and approach is intended to replicate elements on a real world battlefield. The schematic portrays the echelonment of forces on both sides -- friendly and enemy -- with the portals (or gateways) they activate for access to the simulation. The synthetic electronic battlefield we advocate is bounded by the Script and Controller echelon to provide exercise monitoring and control for activities and conditions external to the system's capabilities.

A written script provides the necessary definition for each exercise. It is not a "master events list" but rather a delineation of the tactical conditions. It is provided in sufficient detail to describe the who, what, when, and why. It establishes and baselines external factors that will influence the conduct of the exercise.

A compelling feature of our conceptual architecture is the notional use of autonomous player entities. These can be Intelligent Autonomous Entities (IAE) which have the same fidelity as a manned-simulator (e.g., CCTT). IAE can be organized into units, under the control of ACTORs, to portray higher order behavior compatible with unit level operations. ACTORs are allocated to staff work stations where they represent the subordinate units of the headquarters. DIS protocols are the best near and long term methodology for integrating

across the spectrum of TES systems. Our architecture also supports integration of existing constructive models, such as CSSTSS, CBS, BBS, and JANUS.

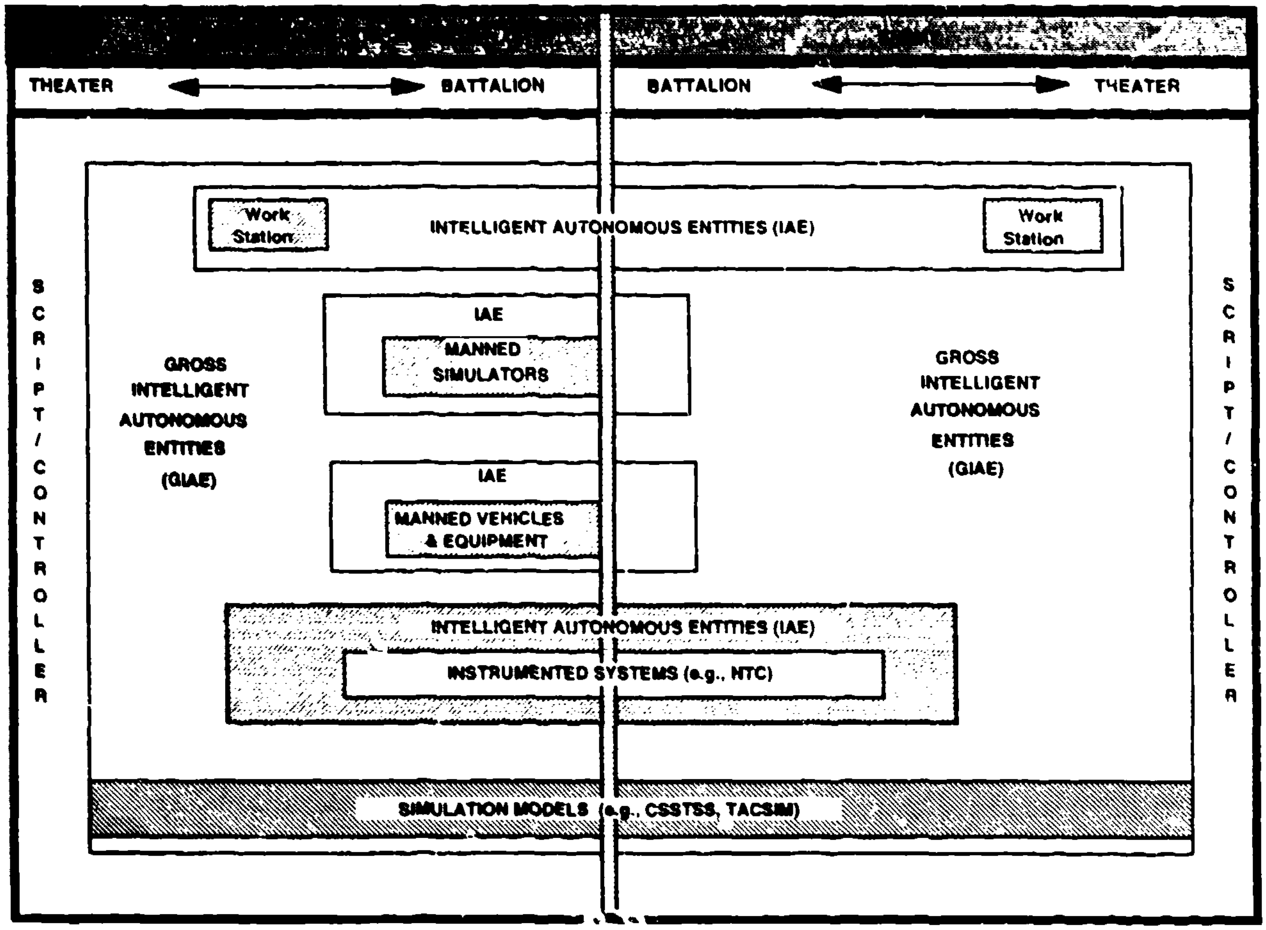


Figure 2. The Synthetic Theater of War

We have offered a conceptual architecture that incorporates many technologies which are evolving and maturing. This paper provides not only a conceptual architecture for integrating TES, but also proposed developmental work in important disciplines along several technical dimensions. We believe that the concepts presented here contribute to a foundation for such an effort. We fully appreciate that important research is ongoing and suggest that a focus along the lines of this paper will offer a high return on investment.

CONCLUSION

The architecture we propose cannot be built without prototyping supported by advanced systems and software engineering. Development cannot be managed as a monolithic system architecture and design which is assembled and then tested. The innovative technical pieces should be developed in parallel, tested using existing government programs and resources, and integrated after they successfully demonstrated.

A BLACKBOARD APPROACH TO COMPUTER GENERATED FORCES

Wesley Braudaway, Ph. D.
IBM Federal Systems Company
Orlando, Florida

ABSTRACT

This paper presents the research for applying the AI blackboard paradigm to the problem of realistically emulating multiple battlefield entities within a training simulator. One challenge of these computer generated forces (CGF) is to emulate human behavior so human controlled and computer controlled entities are virtually indistinguishable. The blackboard paradigm provides a useful framework for attacking the CGF problem. The results of this research demonstrate the usefulness of a blackboard architecture for the CGF problem. The blackboard paradigm provides a means for integrating subtasks of the system that are implemented using different programming paradigms. Also, the "context" and "event" driven control strategy of the blackboard paradigm provides adaptive behavior for the computer control forces. These characteristics discriminate the blackboard architecture from other programming paradigms for use with the CGF problem. The research reported in this paper was funded by STRICOM under contract N61339-92-C-0032.

ABOUT THE AUTHOR

Dr. Wesley Braudaway received his Ph.D. from the Department of Computer Science at Rutgers University in New Jersey while on leave of absence from IBM Federal Systems Company. His thesis research involved Artificial Intelligence Knowledge Compilation and AI Design Automation. Since joining IBM in 1982, Dr. Braudaway has been involved in several research projects applying expert systems to submarine command and control problems. Over the last year he has been the principle investigator of the automated forces research task for applying AI to the computer generated forces problem. Dr. Braudaway is presently part of the Integrated Development Team to develop semi-automated forces for the CCTT program.

A BLACKBOARD APPROACH TO COMPUTER GENERATED FORCES

Wesley Braudaway, Ph. D.
IBM Federal Systems Company
Orlando, Florida

INTRODUCTION

The CGF Problem

A computer generated forces (CGF) system implements semi-automated opposing and ancillary forces that operate on a simulated battlefield and emulate realistic behavior of human lead military units. Training military for combat within the complex and chaotic environment of a full scale battlefield is a difficult and expensive task. Computer simulation coupled with Artificial Intelligence (AI) technologies provide a promising approach for an effective training environment of battlefield combat. This paper describes the research results of combining the AI *blackboard* paradigm[2] and *computer simulation* to automate realistic battlefield units. The research reported in this paper was funded by STRICOM under contract N61339-92-C-0032.

To model realistic battlefield situations, hundreds of entities from friendly and opposing forces must be present. This magnitude necessitates the use of both human controlled and computer controlled entities on the simulated battlefield. One challenge of these computer generated forces is to emulate human behavior so that the human controlled and computer controlled entities are indistinguishable. This implies that the CGF system must not only exhibit realistic behavior in a static environment, but must realistically adapt its behavior to dynamic events and emerging situations occurring on the battlefield. The combination of different events and contexts in which the events occur in a battlefield scenario, can make a conventional control strategy for the CGF solution very complex.

One challenge of extended research is to integrate successful approaches from previous research within a system architecture that also addresses the weaknesses of previous research results. In aggregate, a realistic CGF must be driven by battlefield events and the battlefield contexts in which these events occur. An effective and efficient CGF system must incorporate both knowledge-based (expert system like) decision making for some tasks and algorithmic approaches for other tasks as appropriate for those tasks' requirements. The system must also be easily modified for changes in the operational tactics that it simulates and extendible for incorporating new battlefield entities, tactics, and weapons.

As described in the next section, our preliminary research demonstrated that the AI blackboard paradigm exhibits these characteristics and the research reported in this paper validates this concept by applying the blackboard paradigm to the challenges of the CGF requirements.

This first section summarizes the prototype implementation and the research results. Section two presents an overview of the AI blackboard technology and Section three summarizes the behavior model on which the prototype design is based. The prototype implementation is described in Section four and the research results are discussed in Section five.

The Solution Overview

The objective of our research was to demonstrate the suitability of using the AI blackboard paradigm to integrate both procedural and AI techniques with a simulation system in order to achieve required CGF capabilities. To achieve this objective, we designed and implemented a prototype system that integrates a vehicle motion simulation testbed (IBM's Combined Arms Combat Simulator (CACS)) with a command decision system implemented using the AI blackboard paradigm. This design demonstrates the features of the blackboard paradigm for emulating vehicle behavior and unit cooperation within the bounds of a representative battlefield exercise.

The prototype implements the tactical behavior and coordination of a platoon of vehicles responding to a dynamic battlefield while conducting a single operation. Through this constrained focus, we rapidly prototyped a demonstration that validates the concept of a blackboard CGF solution while having a small impact on research costs and time. Although the scope of this research is constrained, further enhancements can be incorporated to include more complex behaviors to implement a more complete CGF solution.

The prototype system, as shown in Figure 1, is composed of the CACS simulation testbed implemented using the C language, and the "Command Decision System" implemented using the AI blackboard paradigm. The CACS simulates entities (e.g., tanks, helicopters, infantry, etc.) navigating on a plan view terrain map

while avoiding obstacles, following routes, and firing weapons as specified manually.

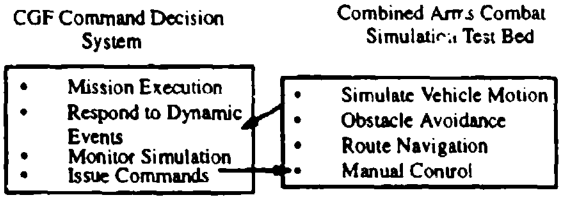


Figure 1 The demonstration prototype high level architecture

Functionally, the command decision system implements a task–oriented, hierarchical behavioral model as described in Le.[3] This behavioral model includes the ability to simulate the collective behavior of combatant agents cooperating toward an objective and the ability to simulate individual agents exhibiting individual behavior in reaction to local terrain and battle conditions. This behavioral model motivates the design of the command decision system as further described in Section 3.

By utilizing this behavioral model, the command decision system can emulate decision processes of some command level (for example; a platoon leader) for some designated mission. The command decision system monitors the events simulated by the CACS, makes decisions based on these observations, refines these decisions into actions, and issues commands to the CACS to implement these action. Because the command system provides automatic control over the CACS, the need for manual control is greatly reduced. To offer flexibility in operator control, however, the capability of overriding the commands from the command system is provided.

The prototype command system is implemented using the "Generic Blackboard Framework" (GBB – Trademark of Blackboard Technology Group, Inc.) which extends the Common Lisp and CLOS (Common Lisp Object System) environment on an IBM RISC/6000. This implementation integrates knowledge sources that utilizes both functional and procedural processes implemented using Common Lisp, rule–based processes implemented using the GBB–OPS language, object–oriented processes implemented using CLOS, and algorithmic processes implemented using the C language.

Summary of Results

The prototype system demonstrates the use and flexibility of the AI blackboard paradigm for the CGF problem. This system utilizes the blackboard features of integrating several different programming methodologies, providing adaptive behavior using a context driven control strategy, and providing an extensible solution for enhancing the CGF system.

BLACKBOARD TECHNOLOGY OVERVIEW

The blackboard architecture is an extension to the classical expert system structure as shown in Figure 2. The blackboard model has three components: a blackboard data structure, a set of knowledge sources, and a controller. Each knowledge source of this architecture is an intelligent agent that can solve some subproblem and uses the blackboard as its communication medium to other knowledge sources.

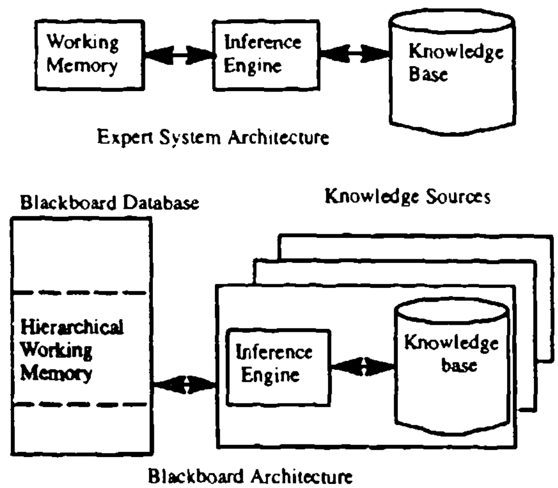


Figure 2 Expert system versus blackboard architecture

The Knowledge Sources

Each knowledge source in the blackboard paradigm is a self–contained specialist of a problem subtask. Each will solve the subtask independently of the other knowledge sources. New knowledge sources can be added to the system without any change to the other knowledge sources.

Each knowledge source is separately implemented as a rule–based program, a logic program, a functional program, or a procedural program using any of the supported languages. A knowledge source contains the knowledge needed to solve some subproblem and the mechanisms for applying that knowledge to the subproblem. During execution, a knowledge source can access information contained on the blackboard and modify the blackboard to record its results. The problem solving control of the blackboard paradigm is dis-

tributed in that each knowledge source is itself responsible for determining if conditions are satisfied that allow it to contribute to the problem solving.

The Controller

The control strategy of the *rule matching – conflict resolution – rule execution* cycle from classical expert systems is relatively complex compared to the control strategy used for the blackboard paradigm. Specifically, the blackboard controller collects all knowledge sources that indicate their readiness to contribute to the solution, adds them to an ordered queue regulated by each knowledge source's evaluation of its own importance, and sequentially gives control to each knowledge source in the queue. In contrast to classical expert systems, a knowledge source — not the controller — decides when it is ready to participate in problem solving and determines the importance of its contribution. These decisions are made by the knowledge source because only it has the knowledge and the means for using that knowledge to determine its own contribution to the solution.

The Blackboard Database

During the problem solving process, the blackboard database is a repository of all information, hypotheses, partial solutions, and problem solving state. Knowledge sources may access the data on the blackboard, remove data from the blackboard, or add new data to this structure. Similar to the working memory of a classical expert system, all communications and interactions between the knowledge sources are achieved through the blackboard database. However, unlike the classical expert system, the data on the blackboard may be represented in many different ways and at different levels of abstraction.

The Blackboard Operation

For this paper, this section describes one operational view of the blackboard paradigm. Many variations of the blackboard paradigm operation exist but will only subtly effect the performance of the CGF blackboard prototype system.

The operation of the blackboard paradigm is illustrated in Figure 3. Activities on the blackboard database such as adding, removing, and modifying information or objects are events that may cause particular knowledge sources to respond. Other special events may be initiated by the blackboard controller or by some knowledge source's execution. For each knowledge source defined for an application is a list of events that cause the knowledge source to respond.

Each knowledge source that responds to a given event during the system's operation, is asked by the controller if it should be activated. A knowledge source's decision to be activated and its evaluation of the importance of its activation are based on the knowledge source's evaluation of this information contained on the blackboard. Thus, the activation of a knowledge source not only depends on the events occurring on the blackboard but also the context in which the events occur.

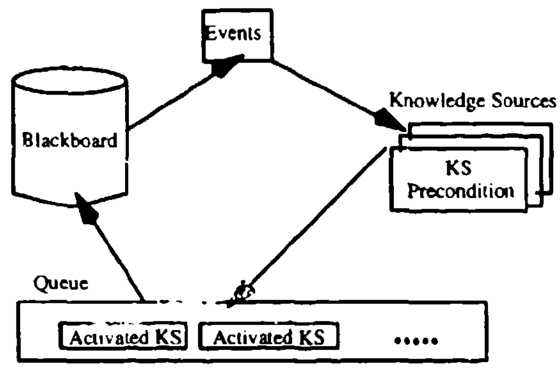


Figure 3 The blackboard system operation

The controller places each activated knowledge source on a queue according to the knowledge source's importance. The first knowledge source on the queue then executes and records its contribution to the solution on the blackboard database. This modification to the blackboard creates new events and the controller cycles until it is told to stop or the queue becomes permanently empty.

Distinguishing Features of the Blackboard Paradigm

Although the blackboard paradigm has some features in common with expert systems and object-oriented programming, there are important features that distinguish the blackboard paradigm for the CGF problem. The blackboard paradigm provides both event and context driven programming while expert systems are more data and goal driven. Although object-oriented programming is event driven where events are actions on objects, its applications are only decomposed in terms of objects and its components. In contrast, the blackboard paradigm allows the integration of both an object decomposition and a functional decomposition of a problem. In addition, for a complex object-oriented application, it can be difficult for the developer to schedule a long sequence of method activations in re-

sponse to an event. In contrast, the blackboard paradigm's controller explicitly schedules the multiple knowledge sources that respond to a single event.

The blackboard paradigm has other advantages over the classical expert system by rectifying weaknesses found in classical expert systems. Each knowledge source can be implemented using the most appropriate paradigm for solving its subproblem. That is, the knowledge source can itself be a rule-based expert system, a logic program, a C language procedure, or a set of Lisp functions. Therefore, the expert system shell's requirement of solving all parts of the problem using the same problem solving method and the same knowledge representation is removed.

This characteristic is particularly useful for solving the CGF problem. A solution must use a variety of both algorithmic and non-algorithmic tasks including mission planning, situation assessment, data fusion, decision making, decision refinement, route planning, monitoring, line of sight computations, and many more. Some of these tasks are best solved procedurally while others suggest a knowledge-based approach.

In addition, the blackboard paradigm allows the cooperative integration of these different problem solving approaches which use and produce data, hypothesis and solution states having different representations and defined on different levels of abstraction. The blackboard paradigm allows these alternative formats to coexist on the blackboard for use during problem solving.

The blackboard paradigm's event and context driven control strategy is also a requirement of the CGF system. By representing the simulated battlefield on the blackboard, any events occurring on the battlefield correspond to events occurring on the blackboard. Each knowledge source reacting to these events can react to the context on the battlefield in which these events occur since this context is represented on the blackboard. The correspondence between the operation of the blackboard and the cognitive unit behavior to battlefield situations will provide the ability to simulate units adaptively reacting to the dynamic battlefield.

THE CGF BEHAVIORAL EMULATION

Generalized Combat Model

The goal of modeling automated forces is to develop both the ability to simulate collective behavior of combat units and the individual combat agents' behavior of reacting to local terrain and battle conditions. The agents defined by this model must be able to mimic the cognitive capabilities of their human counterparts by perceiving their environment, updating and maintaining a model of the developing tactical situation, planning actions, reacting to dynamic situations, monitoring their execution, and communicating to other agents. Also, to be effective, this generalized model must not only characterize the behavior of a particular type of combat agent but also specify a functional model whose instantiation can be used to simulate a variety of combatants including tanks, infantry, air support, and so on.

As shown in Figure 4, the major functional elements of this model include basic vehicle movement and maintenance, obstacle avoidance, unit motion coordination, military mission/task execution, and intelligence/planning. These elements are arranged hierarchically in a manner similar to a subsumptive approach.[1] However, the difficulty of using a purely subsumptive approach is that it only solves the reactive role of the combat agents (i.e., event driven) whereas in a complex battlefield environment, acceptable behavior depends on the battlefield and the operational context in which the agents must react.

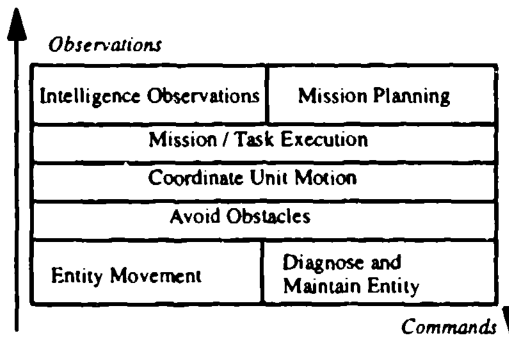


Figure 4 The generalized combat model

To achieve this capability, the hierarchical model must be adapted to include a task-oriented approach similar to the one used by Schudy and Duarte[4]. Therefore, behavior at each hierarchical level must be defined in terms of not only the type of agents but also the type of battlefield operations conducted by these agents.

This behavioral model is also based on the conventional communication paths between cooperative entities and command levels. Within the military command echelon, commands are disseminated "down" the chain of command. Plans from the highest level are refined into tasks and then into actions as they are passed to lower levels in the echelon. The combined behavior achieved through these actions will implement the original plan. At the higher levels in the echelon, information is abstracted from observations occurring at lower levels

and is utilized to construct the mission plans. By accumulating observations "up" and passing plans "down" the command echelon, the entities will cooperate to achieve a mission objective.

The Battlefield Exercise

Before designing and implementing the prototype system for this research effort, we defined a set of operations and behaviors that the prototype's units would emulate. These requirements are specified within a battlefield exercise that bounds the set of expected events and the set of expected behaviors to a manageable set for the research activity.

The command system designed and implemented for this research task emulates the behavior of multiple platoons conducting independent zone reconnaissance operations. Opposing forces are controlled manually by an operator to create various scenarios that test and demonstrate the automated platoons' responses to dynamic events. The *operation order* requires each platoon to conduct a zone reconnaissance in a bounded zone containing three phase lines, coordinate progress at these phase lines, and return from the third phase line to the line of departure via the most expeditious route. The purpose of the reconnaissance mission is to detect, classify, and react to any opposing forces sighted from within the reconnaissance zone but not to engage these opposing forces.

PROTOTYPE SYSTEM ARCHITECTURE

The behavioral model for the command echelon is used to design the command system that emulates the performance of a cooperative group of entities (e.g., a platoon). As described in Figure 1, the system architecture is composed of two processes: the CACS and the command system.

The CACS was not developed prior to this research effort. From an AI perspective, the route following and obstacle avoidance techniques implemented in the CACS are interesting and are reported by Le.[3]

Blackboard Organization

To emulate the platoon units on the battlefield, the command decision system must be able to observe events occurring on the battlefield, determine effective responses to those events, refine those responses into vehicle actions, and communicate those actions to the CACS.

The command system is organized using abstraction levels on the blackboard. These levels and the knowledge sources that interact with each level are abstractly represented in Figure 5. Information represented at a particular level of the blackboard corresponds to the information needed by an associated level of the military command echelon. For example, a vehicle's commander must be aware of his route to an identified objective while a platoon leader must be aware of the area where the platoon is actively operating.

The blackboard database is divided into two primary levels: the command level and the simulation supervisor level. The command level contains all information affecting the command decisions for the emulated units. The simulation supervisor level contains all battlefield information obtained from the CACS and commands sent to the CACS. The command level is further divided into levels corresponding to levels of a military echelon (for example, battalions, companies, platoons, and platforms). For the implemented prototype, the command level contains the platoon and platform levels.

The command level of the blackboard contains the organization of each unit controlled by the command system and any information affecting that control. For example, objects at this level describe a platoon's organization as composed of specific platforms (i.e., tanks, infantry, armed fighting vehicles, etc.).

The simulation supervisor level of the blackboard contains information about the simulated battlefield and every vehicle on the battlefield as simulated by the CACS. Therefore, the simulation supervisor level contains perfect information about the battlefield and activities occurring on the battlefield. To accurately emulate the performance of manned vehicles, the decisions made at the command level should only be based on information realistically available to the manned vehicles. For example, vehicles that are more than 3500 meters from the controlled units cannot realistically be seen by those units. Therefore, command decisions for these units should not be based on the existence of these vehicles.

This motivates the design decision of dividing the blackboard into the command level and the simulation supervisor level. The simulation supervisor level serves to *hide* information that the controlled units cannot realistically utilize. There are no features of the blackboard COTS product used for this study that provides secured access between these blackboard levels. Therefore, information hiding is achieved through design alternatives and programming discipline. For this prototype system, the knowledge sources at the simula-

tion supervisor level choose the information that should be available to the command level and *filters* this information to the command level of the blackboard.

Knowledge Source Responsibilities

As shown in Figure 5, knowledge sources are associated with a particular level of the blackboard. However, knowledge sources can abstract or refine information from one level of the blackboard to another level. For example, the "Filter Information" knowledge source moves entity detection information to the Command level only if a controlled entity can make that detection.

Simulation Supervisor Level: The set of knowledge sources at the supervisor level accomplish three tasks. One group of knowledge sources monitors the status of the battlefield and the entities simulated on the battlefield. Another group sends commands to the CACS that affect the entities simulated on the battlefield. The last group of knowledge sources filter information from this blackboard level to the command level based on the ability of the emulated units to realistically perceive this information. The knowledge sources at the supervisor level are either algorithmic or functional processes as represented in Figure 5 as processes defined by rectangular boxes.

Platform Command Level. The platform sublevel is the lowest level of the command system and contains information about each of the controlled platforms. The knowledge sources at this level specifically control a single platform as an independent unit. Since most of the platform dynamics are implemented as part of the CACS, the knowledge sources emulating a platform in the current prototype provide the automatic routing and communications capabilities of each platform.

Platoon Command Level. The platoon sublevel of the blackboard contains information about the organization of the platoon and the platoon's current operation. For this prototype system, the knowledge sources at this level emulate a zone reconnaissance operation for a platoon of M3 calvary fighting vehicles. The activities implemented by the knowledge source include coordinating the platoon at the zone's phase lines, reacting to opposing force detections and classifications, and reacting to opposing force attacks.

The "M3 Platoon Recon" knowledge source is a rule-based program, as indicated in Figure 5 by the ellipse, that emulates the decisions of a platoon leader responding to battlefield events. For the reconnaissance mission, these events occur when a platform reaches a control measure route way point, when a platform detects or classifies an opposing force, and when a platform detects a munitions firing. The M3 platoon reconnaissance knowledge source is responsible for creating the platoon's response to the battlefield events within the parameters and tactics defined for the platoon's mission. In general, a knowledge source at this level is responsible for emulating a specific type of platoon conducting a specific operation.

Design Summary

Every object on the command level of the blackboard refers to a particular vehicle, platform, or platoon leader. This includes the opposing force detections, firing detections, and control measure way points. Each knowledge source activation is created in response to some event affecting these objects and is therefore

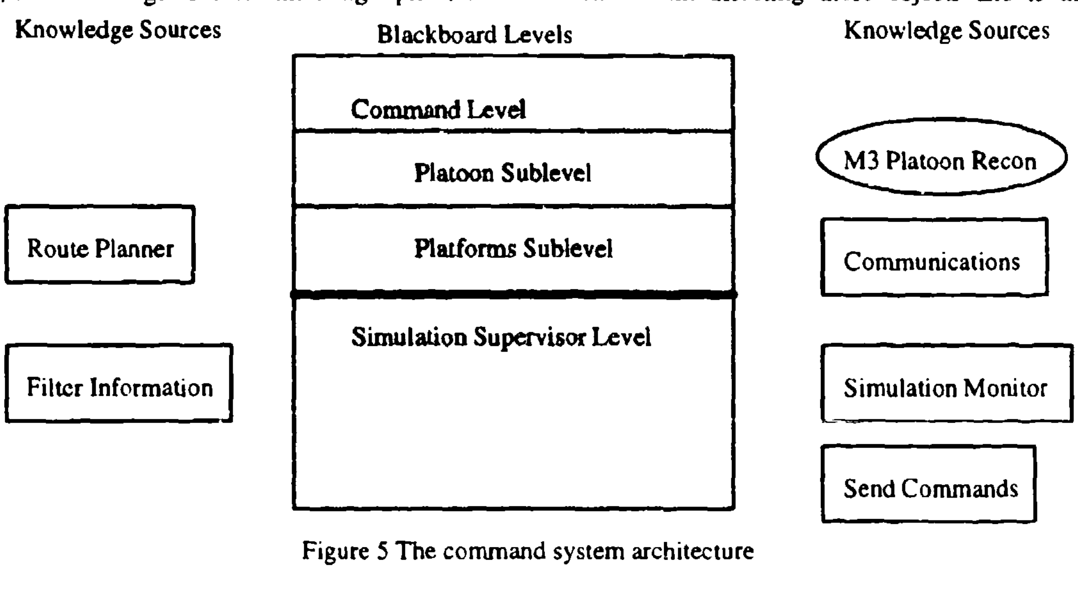


Figure 5 The command system architecture

associated with a single platoon. Thus, with no additional code and utilizing the same blackboard, these knowledge sources can control multiple independent platoons without incurring any interactions between these multiple platoons. For example, multiple M3 platoons, each conducting reconnaissance operations, will behave independently. While one platoon is responding to a firing detection, a second platoon will continue its reconnaissance operation without any influence from the activity of the first platoon.

RESULTS AND CONCLUSIONS

The automated control of forces for the CGF problem is similar to the classic Command, Control and Communications (C3) problem. In its general form the command and control components for CGF should have a hierarchical structure that matches a military echelon. The highest level of command and control must process and fuse data from different sources, assess the battlefield situation based on that data, plan mission objectives to react to the battlefield situation, and allocate the resources required to achieve those objectives. The next levels in the command and control model for CGF must progressively refine these objectives into tasks and then into actions at the unit level on the battlefield. The lower levels must also pass abstractions of their battlefield observations up the hierarchy to aid the decision making processes at these higher levels.

Therefore, to effectively emulate a C3 model, the system should include tasks such as data fusion, situation assessment, planning, decision making, task refinement, action execution, battlefield monitoring, detection, classification, routing, and others. However, the decision process for each unit must be able to "opportunistically" respond to events occurring on the battlefield within the parameters defined by doctrine and the current situation (i.e., the context).

The results of research reported in this paper show that a blackboard paradigm can provide the control of the computer generated forces using a C3 model. To evaluate the concept of using the blackboard paradigm for the CGF problem, a blackboard CGF prototype was designed and implemented to demonstrate the relevant blackboard features including its integration of different programming methodologies, its event and context driven control strategy, and its extendibility for the CGF problem.

Programming Paradigm Integration

The implementations of all tasks defined by the Command, Control and Communication model are not well suited to a single programming method. That is, some C3 tasks are best solved algorithmically such as computing the line of sight between two battlefield entities. Other tasks such as situation assessment are best solved using a knowledge-based approach. The variety of processes defined for C3 requires the integration of several problem solving methodologies into a cooperative system.

As provided by the blackboard paradigm, the prototype system utilizes the ability to integrate multiple programming paradigms. This allows each subtask to be implemented using an appropriate programming method. For example, decision making involved in reacting to firing events, opposing force detection events, and platoon coordinate events were implemented using rule-based programming. Likewise, the line of sight and routing algorithms that require fast numeric computations were implemented as conventional, compiled procedures.

By utilizing the blackboard paradigm's common communication medium (the blackboard) and its control strategy that is distributed among the independent knowledge sources (each implemented using the most appropriate methodology), these alternative programming methods were easily integrated to contribute to the CGF solution. This flexibility allows the design of each CGF component to match a cognitive model for the operation of actual battlefield forces.

Context and Event Driven Behavior

At any instant during the battlefield simulation, events may arise that require a response from the emulated entities on the battlefield. These events can include an entity's arrival at some key terrain location (e.g., an assembly area), a detection of some opposing force, a detection of a munition firing, the arrival of some key time event, and many others. With no other complexities, an event driven simulation system could adequately cope with this requirement. However, in the CGF simulation many more variables are introduced.

The variety of events that must be handled by the control system and the variety of responses available with respect to the battlefield context, greatly increases the complexity of an event driven control strategy for the simulation. That is, the different combinations of events, contexts, and sequences in which the events occur represent a combinatorial control problem. Prede-

fined sequential control strategies (e.g., procedural or finite state automata approaches) for this problem will not only be very complex, but will also provide a fixed (or static) behavior to an event because they cannot account for all contexts and sequences in which the event occurs. To provide adaptive behavior, the system must be able to easily and correctly respond to battlefield events within the context of the battlefield.

The blackboard data structure in the prototype system abstractly represents the simulated battlefield including terrain entities, units, detections, and so on. Events on the battlefield correspond to events on the blackboard which activate particular knowledge sources. In response to these events, the knowledge sources individually evaluate the battlefield context represented on the blackboard to determine if they should respond to not only the event but also the context in which the event has occurred. Therefore, by using the context and event driven control strategy provided by the blackboard paradigm, the control strategy of the prototype is the simple control loop described in Section 2. Also by using this control strategy, the demonstration prototype adaptively responds to the dynamic battlefield events.

Extendible CGF Design

A CGF system can have many uses including training, evaluating battlefield tactics, and evaluating the effectiveness of new units or weapons. To provide capabilities for these uses, the system must be designed to allow the incorporation of new unit types, new weapon types, and new battlefield tactics. That is, the system must be modular and easily extendible.

By implementing separate subtasks as knowledge sources that are independent and interact with other subtasks only through a blackboard, the blackboard paradigm provides the means for making the system very modular and extendible for added units, tactics, and operations. By encoding specific subtasks into separate, independent knowledge sources, the system can be functionally extended by adding new knowledge sources. These knowledge sources can be designed by acquiring only the expertise that is relevant to the new task rather than expertise about the entire CGF problem. For example, to incorporate a new platform type we need only interview an expert on the tactics and capabilities of that new platform.

To construct a system that is easily extendible and modifiable, the CGF system architecture should conceptually model the architecture and behavioral model of the emulated military echelon. The hierarchical nature of the blackboard data structure provides a skeleton upon which this behavioral model can be constructed. The blackboard can be divided into levels which segregate and abstract the data and decisions among the appropriate command levels. The independent knowledge sources can provide the appropriate behavior for each entity on each level of the military echelon.

This demonstration prototype can easily be extended both horizontally, providing new platoon unit types and operations, and vertically, providing the higher command levels of company, battalion, and so on. The knowledge sources associated which each level can be responsible for making decisions relevant to that level, refining decisions from high levels into actions, or abstracting observations from lower levels. The command echelon, the decision and mission refinement down the echelon, and observation abstraction up the echelon are all explicitly represented within this blackboard design. This provides a common model of battlefield command operation that should enhance the extensibility and modifiability of the system.

The Command System Implementation

The prototype command system (not including the CACS) contains approximately 1,100 lines of C code. This code implements the computation intensive tasks (e.g., route planning) and the system's communication with the CACS. The system also contains approximately 4,200 lines of Lisp code. This code specifies much of the blackboard and knowledge source definitions, and implements the basic blackboard manipulation functions such as filtering information on the blackboard. The "M3 platoon reconnaissance" knowledge source contains 38 rules and 400 lines of supporting Lisp code. The command system also defines 30 different types of objects that can be created on the blackboard for this implementation.

SUMMARY

The research task described in this paper addressed the requirement for an architecture that allows the integration of various CGF subtasks, provides a realistic emulation of multiple battlefield entities, and allows the extensibility required of a CGF solution. The AI blackboard paradigm offers features that satisfy these requirements. The research confirms this hypothesis by demonstrating a blackboard CGF prototype system that emulates multiple, independent M3 platoons conducting zone reconnaissance operations.

REFERENCES

1. Brooks, R., A robot layered control system for a mobile robot. *IEEE Journal of Robots and Automation*, RA-1(1), pages 1-10, 1986.

2. Engelmore, R. and Morgan, A., editors. *Blackboard System*. Addison Wesley Publishing, California, 1988.

3. Le, H. T., Phinney, S. E., and Seward, V. C. Semiautomated forces: A behavioral modeling approach. In *The Proceedings of the Thirteenth Interservice/Industry Training Systems Conference*, December 1991.

4. Shudy, R. B., and Duarte, C. N. Advanced autonomous underwater vehicle software architecture. In *The Proceedings of the Symposium on Autonomous Underwater Vehicle Technology - AUV'90*. Washington, D.C.,1990, pages 9-22.

ENERGY LEVEL MODELING: A NEW APPROACH TO REAL-TIME ECM RADAR THREAT SIMULATION

Drew Tucker, SBS Engineering, Plano, TX
Lt. Kurt S. Collom, U. S. Navy, VF-124, San Diego, CA

ABSTRACT

Effects level modeling in radar simulation has been the traditional approach for satisfying Electronic Countermeasure (ECM) training requirements. A new Radar Environment Simulator (RES), developed for a U. S. Navy F-14A Weapon Systems Trainer, utilizes design principles which go beyond the traditional. The jammer models in the RES are based on detailed modeling of real-world transmitted and received energy levels ("energy level modeling"). This design approach is used instead of simply attempting to duplicate visual effects ("effects level modeling"). While either of these methods can provide an accurate simulation under normal operating conditions, the energy level model has significant advantages when ECM is introduced into the scenario. The result is a trainer that is more realistic in its response to a large set of radar operator actions and threat variables.

Energy level modeling can be applied to the simulation of systems designed for the detection, acquisition, and tracking of various targets. This design principle enables the software to emulate all radar system behavior without anticipating each unique scenario. In addition, non-standard radar operator inputs to an actual radar system interface are processed real-time using a detailed radar model allowing realism never before possible. Consequently, the goal of preparing a trainee for a wide variety of ECM threats and threat signatures is achieved to an extent not feasible through traditional effects level simulation.

ABOUT THE AUTHORS

Drew Tucker is a software engineer with SBS Sensor Systems, a division of SBS Engineering, which was formerly part of Merit Technology. He has been designing software solutions for use in aircraft trainers and simulators for seven years. In addition to the F-14 trainer, he has developed real-time systems for AV-8, F-16, F-22, and B-52 simulators. Mr. Tucker holds a B. A. in Mathematics and Mathematical Science from Rice University and an M. S. in Computer Science from Southern Methodist University.

Lieutenant Kurt S. Collom is an F-14 Instructor RIO (Radar Intercept Officer) currently assigned to VF-124, the West Coast F-14 Fleet Replacement Squadron. He has over 1500 hours in all models of the F-14 and has been instructing for almost two years following three years of deployed F-14 operations onboard USS Nimitz (CVN-68). Lt. Collom graduated from the United States Naval Academy with a B. S. in Aerospace Engineering.

ENERGY LEVEL MODELING: A NEW APPROACH TO REAL-TIME ECM RADAR THREAT SIMULATION

Drew Tucker, SBS Engineering, Plano, TX
Lt. Kurt S. Collom, U. S. Navy, VF-124, San Diego, CA

INTRODUCTION

Most aircraft, ships, and missile systems in the military inventory utilize radar as one of the primary sensors in the performance of their mission. Simulators and trainers based on these systems employ radar simulation for training in the operation of these increasingly complex radars. Effects level modeling is the predominant method that has been employed in radar simulation. This traditional method involves designing simulation algorithms based on resultant effects rather than the underlying processes.

A new approach was used when developing a radar environment simulator for the U.S. Navy's 2F112 F-14A Weapon System Trainer at NAS Miramar. This approach, termed "energy level modeling", allows for a more intuitive design, a simpler integration, and expanded trainer capability. The merits of this method will be discussed in terms of ease of development and suitability for training. The focus will be on energy level modeling algorithms used for simulation of electronic countermeasures (ECM), or "jamming".

HISTORICAL DESIGN APPROACH

There are two basic approaches to the problem of providing a radar simulation. One is an effects level model where the emphasis is on providing the correct display appearance. The second uses energy level modeling to track the emission of microwave energy, its interaction with the environment, energy captured by the antenna, and hardware-induced modifications to the resulting signals. Under normal conditions, either approach can provide an accurate simulation of the radar.

Historically, effects level modeling has been chosen for sensor simulation of ECM. Until recently, this simplistic approach was all that could be implemented for reasonable cost in a real-time system. Although simulations were developed using energy level principles, these systems did not have real-time capability and were not designed with training in mind.

Hardware Platform Selection

While not directly related to the design methodology, another distinction can be drawn between two historical approaches. In the past, the substantial processing necessary for real-time radar simulation dictated designs utilizing specialized analog hardware or custom digital processors. However, with the increasing availability of low cost, high speed processors, a software design can now be implemented. The approach chosen for the F-14A trainer upgrade was to use commercial, off the shelf (COTS) digital processors. This allowed for the radar simulation algorithms to be purely software-based, allowing maximum flexibility. For reasons of cost, implementing a software design on low maintenance COTS equipment is preferable to designing and maintaining simulator-unique hardware.

Effects Level Simulation

Effects level design concentrates on devising an algorithm which presents the appropriate display to the operator. The algorithm is primarily concerned with the appearance of a system capability or artifact and is generally very efficient. This approach can provide an accurate simulation of a radar during its standard modes of operation. The requirements analysis and system engineering is done empirically. For instance, the detection ranges of different types of targets are determined by performing a limited (and hopefully representative) set of experiments using the sensor device to be simulated. The resulting "rules" are encoded in the hardware or software which implements the model. Different aspects of the radar being simulated can be modeled and modified in isolation as the system is developed and integrated. Changing the characteristics of one artifact will have no effect on another artifact. This makes adjustment of environmental effects

(like reducing atmospheric attenuation) a difficult proposition.

The traditional effects level approach to ECM simulation provides only the visible display effects of a jammer without performing any detailed modeling of the energy responsible for these display effects. Modeling the brightness of a jammer on a radar display as a direct function of its distance is a simplified example of this approach. When ECM is modeled in an effects level simulation, it can become isolated from other models in the simulation such as targets and landmass. Often, the effect of radar operator actions is not fully taken into consideration.

The effects level design is based on the philosophy that for every environmental or operator action there is some related effect that may appear on the displayed output. An accurate but tedious extension of this approach would be to catalogue every possible combination of conditions and their corresponding outcomes. This endeavor would produce an exhaustive catalogue. Writing the simulation software, however, would call for little actual design. It would instead require a tedious data entry effort. Computer hardware to handle such a program in real-time would have to be generations beyond what is available today.

Since such an accurate simulation is not feasible in real-time, the effects level compromise is to simplify the "real world" cause and effect table down to a few of the most meaningful relationships. This provides a simulation that gives a reasonable approximation to reality, while attempting to provide the greatest fidelity in areas of interest to training. When a limited number of effects are simulated, the computer algorithm becomes quite efficient. This approach allows software models to be easily constructed once accurate information is collected about causal connections and training priorities. Testing is simplified because the limitations of the design can be identified from the beginning. These features are some of the reasons that effects level models became the standard in radar simulation.

On the other hand, there are several drawbacks to the effects level approach. Without access to direct experience in every complex training scenario, it may be more difficult to identify design requirements for an accurate implementation. If requirements are changed or clarified, additional engineering is required due to lack of design flexibility.

The impact on training is significant. A simulator designed with effects level principles tends to be more generic and less flexible. Training scenarios tend to be predictable. The system may be less responsive to a student's or instructor's input. As a consequence, the trainer tends to be less realistic in operation, and training goals are not fully realized.

ENERGY LEVEL MODELING - A DESIGN PERSPECTIVE

The alternative to effects level simulation is an approach called "energy level modeling". The goal of a faithful simulation with the correct effects is the same, but the solution is a bit more complicated. The focus is on understanding and modeling the underlying physical principles that bring about a sensor effect rather than on the effect itself.

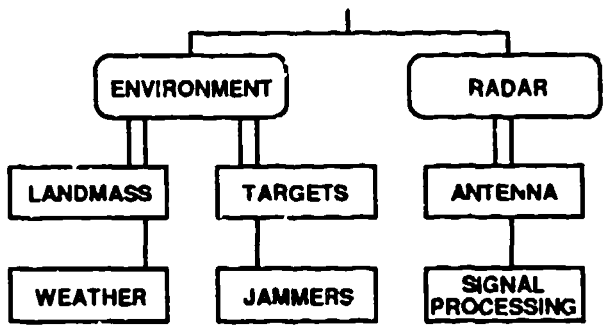


Figure 1 Object Hierarchy

In sensor simulation, the primary physical principle characterizing the model is energy level. This is true whether the sensor being modeled is designed to react to radar, infrared, light, or even sonic energy. The aim of energy level modeling is to determine the strength of a return on a display (or to send to another subsystem) by finding the amount of energy present at several significant points in the simulated environment. Although the name "energy level modeling" was chosen based on the principle of tracking energy intensity throughout a model, there are other characteristics of the signal besides energy intensity, such as wavelength, phase, coherence, spectrum, and pulse repetition frequency (PRF) that are tracked in the same manner.

Design Details

Although not inherent to the principle of energy level modeling, an object oriented design (OOD) strategy is recommended for implementation of a sensor simulation. Object oriented design allows for a logical breakdown of the components of a simulation. It allows for data encapsulation, while still providing a mechanism to share data between objects.

One example of an object oriented breakdown for a sensor simulation begins with the distinction between environment simulation and signal processing (see figure 1). Separate software routines simulate the real world outside the sensor and the processing of received energy inside the sensor. The environmental routines can be further

broken down into objects that relate to landmass, weather, target, and jammer simulation.

Overview - In the case of a radar simulation, the first point at which energy is measured is at the radar transmitter. This is accomplished by performing a computation based on transmitter power, frequency, modes, and any simulated transmitter malfunctions. The effect of most operator actions is taken into account at this point. Next, the percentage of transmitted energy that arrives at the radar reflector is computed based on atmospheric and range attenuation and antenna gain pattern. Using the characteristics of each simulated environmental reflector, the energy returned toward the sensor is calculated. Finally, the amount of that energy is reduced by the return-trip atmospheric and range attenuation, and the receiver antenna gain is factored in. Once the intensity of radiation received in the waveguide is computed, it can be summed together with received intensities of similar character from other environmental reflectors. The appropriate signal processing computations may then be performed in order to determine whether the energy meets the required thresholds for display or detection. It is during this final step that other operator actions such as manual gain or threshold adjustment come into play.

Landmass - The terrain return is computed by sampling a local area map based on Digital Mapping Agency (DMA) data. This provides the altitude and reflectivity of a representative sample of points in the landmass database. At any given point, the return from a terrain patch is computed based on its reflectivity, distance, and angle of incidence. Because landmass is simulated as area clutter, the computed value for received power is attenuated in proportion to the cube of the range to the terrain reflector. The effect of the landmass blocking the line of sight to other environmental reflectors is also taken into account.

Weather - The weather return is computed based on the intensity of the weather cell and distance to the cell. Weather also has the effect of partially attenuating targets and landmass whose return must pass through the weather cell. Each portion a weather cell can also attenuate return from other portions behind it, so weather is simulated as volumetric clutter. The degree of this attenuation will vary based on weather intensity and (in the case of a radar sensor) transmitter frequency.

Targets - Target return is calculated from distance to the target, its radar cross section (RCS), and any atmospheric or weather-related attenuation. The received power is attenuated in proportion to the fourth power of the range to the target. This is done because, unlike landmass, the amount of reflecting target area does not vary with range. Objects which are smaller in size than the radar resolution are simulated as point targets. Larger targets can be combined with landmass or split into multiple point targets.

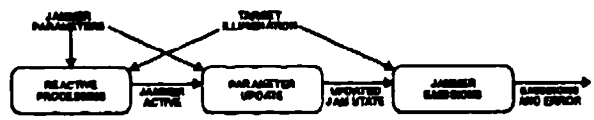


Figure 2 Jammer Data Flow Diagram

Jammers - When simulating the radar return of ECM, different strategies may need to be used based on the jammer type. Two examples are given here.

A simple noise jammer will emit high levels of radar energy regardless of the presence of other emitters. Model calculation for this threat begins at the jammer itself. Its effective power is attenuated by its own antenna gain. At this point the energy is treated as if it were a reflected radar return. Its one-way range attenuation and receiver antenna attenuation are calculated and applied to the energy generated.

A coherent repeating or transponding jammer echoes back an amplified or modulated version of any radar energy it may detect that falls within a certain frequency or power range. This type is handled in the same way as a target return. The difference is that the repeater gain is used instead of RCS to figure the turnaround differential at the target. In this way practical limits on total jammer power output can be simulated.

All jammer types can be simulated using a similar object structure. An example of a typical jammer object data flow is shown in Figure 2.

Signal Processing - In the signal processing routines, the return from each of the environmental objects is summed into one total return. Antenna gain is taken into account. Inputs from the sensor operator, like mode or channel switching, are factored in. Internal controls such as automatic gain control are applied.

Analysis

A limitation of this approach is that a greater investment in research and requirements analysis is required. This additional research is needed to identify sufficient underlying radar and ECM characteristics to provide an accurate energy level model. Feedback from experienced sensor operators remains an important part of the design process. An energy level model may also consume more processor time than a simple effects level model.

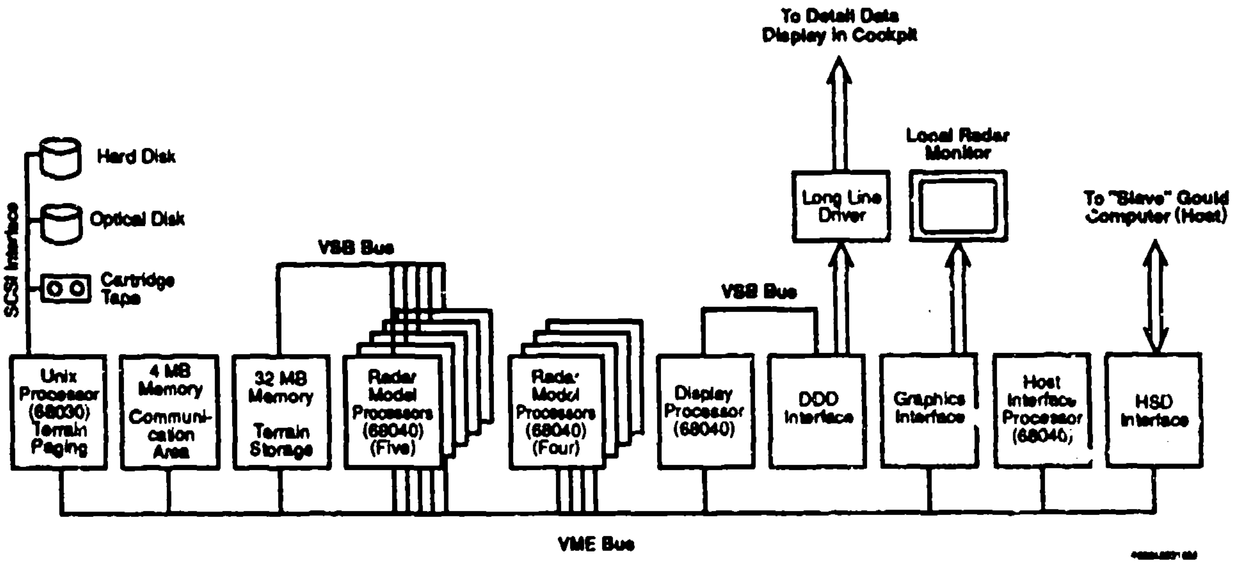


Figure 3 RES Processor Architecture

This extra investment during requirements identification provides a benefit during the design of the actual algorithms. This is a good example of allowing the computer to do the work rather than the programmer. With the proper radar equations encoded into the software, maintainability is improved. Enhancements can be implemented without additional research into the underlying radar equations.

ENERGY LEVEL MODELING - A TRAINING PERSPECTIVE

There are several benefits of simulator training utilizing a real-time energy level model instead of the simpler effects level model. The most predominant of these is the improved realism of the simulation afforded the student by providing an interface with the actual radar system software and control devices. This is preferable to using a software engineer's conception of radar displays, especially those affected by ECM, for two reasons. Pre-designed displays may not necessarily be correct for a given scenario, and planning for every possible scenario creates an inflexible training environment.

Realism is further enhanced by simulating the inconstant nature of energy waveform interactions emanating from two or more jamming sources. In the real world these waveform additions and cancellations would have to be interpreted by the radar system and displayed in a form consistent with running radar software. Depending on the various levels of the received energies, atmospheric attenuation, operator selected gains and a host of other variables, actual displays will be constantly changing. This phenomenon is not present at all in the effects level model. While the predictability of effects level modeling allows a specific scenario to be repeated unchanged until the desired response is elicited, the scenario itself is not a true simulation.

Another benefit realized by an energy level modeling solution is that a wider variety of potential effects are available for demonstration. By allowing the instructor to adjust ECM parameters such as sweep rate, frequency bandwidth and repetition rate to real or suspected values, an unlimited number of training scenarios can be presented with highly accurate displays. The energy level model is completely adaptable to future generations of jamming platforms with little additional software coding required.

Furthermore, all effects are interactive. The programmer does not have to design displays to fit all possible scenarios. All combinations of jammer energy are simply calculated, summed and passed to actual radar software for display. As a result, a wide variety of jamming platforms is available for display. This allows for the utmost in flexible training opportunities.

Given that real-time operator action is factored into the equation as the scenario proceeds, the display effects resulting from operator input are realistic and instantaneous. This allows the instructor to reinforce proper decisions "on-the-spot" which equates to both a better understanding of ECM cause and effect concepts and a faster learning of correct procedures and techniques. Consequently, while effects level models afford limited training for a finite number of scenarios, the realism and flexibility offered by the energy level model allows a greater amount of quality training to be conducted.

APPLICATIONS

Energy level modeling is a relatively sophisticated approach to sensor simulation. Recent improvements in the power and cost of general purpose microprocessor-based computers make approaches such as this one possible. The suitability of energy level modeling to ECM simulation has been demonstrated by the F-14A Weapon Systems Trainer (WST) upgrade. Also, there are numerous other applications where this approach would be appropriate.

F-14A Weapon Systems Trainer

The main goal of the F-14A WST upgrade was to provide an accurate and maintainable ECM simulation at a reasonable cost. Rather than modify the existing obsolete radar simulation equipment, a new Radar Environment System (RES) was built. The choice of real-time computational hardware, a VME chassis with eleven 68040-based Motorola MVME-165 cards (see Figure 3), helped make the energy level model a success on this upgrade. These processors provided the power to make energy level computations using floating point arithmetic in real-time.

An extensive list of electronic countermeasures was provided on the new RES. Some of the ECM simulated were spot noise jammers, barrage noise jammers, velocity gate stealers, range gate stealers, false doppler target generators, repeater noise jammers, cross polarization jammers, swept amplitude modulators, and combinations thereof. In all, over thirty distinct jammers were simulated.

The RES simulated the functionality of the AWG-9 radar subsystem (see Figure 4).

Accordingly, it was required to interface with actual on-board computer (OBC) subsystem hardware and software by way of the host computer (see Figure 5). As a result, the ECM simulation was required to generate output that was realistic enough for identification by threat recognition algorithms in the OBC. An accurate presentation on the radar display was also a requirement. The energy level approach proved to be ideal for meeting these requirements, especially when simulating scenarios involving target screening and standoff jamming.

After the RES was successfully installed, an additional software upgrade was delivered. The schedule and cost of this modification would not have been possible without the flexibility of the energy level modeling design approach. Furthermore, such a change would have been prohibitive to make in the original analog hardware radar simulation system that was replaced by the RES.

Potential for Other Systems

Such an approach has many other potential applications. Any trainer or simulator that uses analog equipment or older digital equipment for its sensor simulation subsystem could be a candidate for such an update. Today's digital systems have the power to allow engineers flexibility in their software design. Designers can concentrate more on the radar models and spend less time worrying about software shortcuts (such as using integer arithmetic or assembly code) to satisfy real-time requirements.

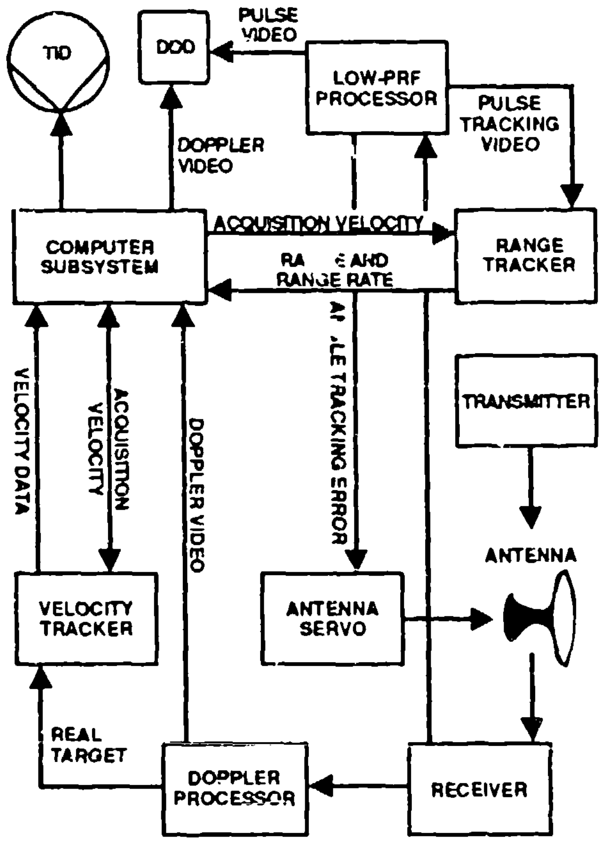


Figure 4 AWG-9 Radar Subsystem

This approach also makes sense for any system where improved fidelity or realism is crucial to the training objective. If the simulation host processor is sufficiently fast and has enough spare capacity, an energy level sensor model could be added in software with no hardware modifications at all. The improvements in microprocessor technology have made feasible new software design approaches that were not possible even five years ago.

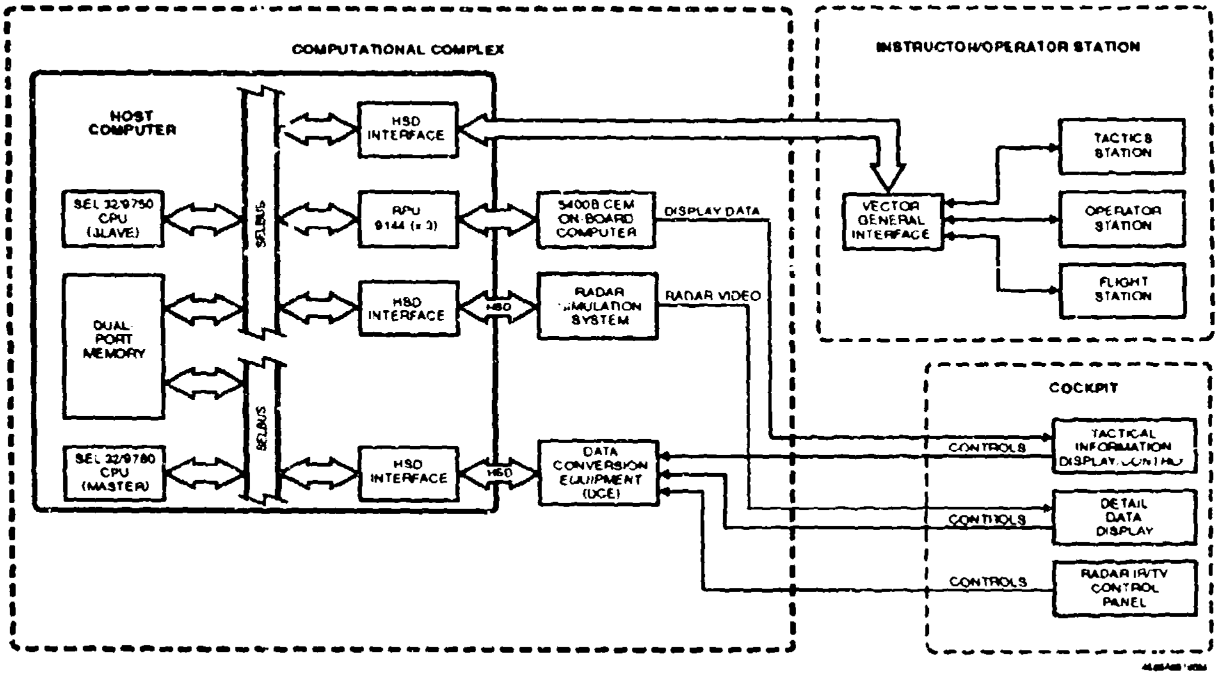


Figure 5 Trainer Diagram

CONCLUSION

The software design approach called energy level modeling, made possible by recent advances in hardware technology, has many benefits from both a design and training standpoint. Reliability, flexibility, and fidelity are all enhanced. It is hoped that others will be able to apply these principles to a wide range of real-time applications. As computational power increases, software design solutions must continue to evolve to exploit the potential of machines providing greater speed and capacity. The principles that underlie energy level modeling should inspire continual methodology improvements as hardware capability grows.

ELECTRONIC COMBAT SIMULATION IN A NETWORKED, FULL MISSION REHEARSAL, MULTI-SIMULATOR ENVIRONMENT

David W. Galloway, Patrick G. Heffernan, E. Allen Nuss, Charles M. Summers
TRW Avionics & Surveillance Group, Warner Robins Avionics Laboratory
Warner Robins, Georgia

ABSTRACT

The Integrated Electronic Combat Simulation System (IECSS) has been developed for the MH-53J and MH-60G Weapon System Trainers (WSTs) and is under development for the HC-130P Aircrew Training Device (ATD). This system provides dynamic simulation of the closed loop Electronic Combat (EC) environment to support multiship operations for eight networked training systems. The IECSS simulates: (1) The electromagnetic and infrared environment; (2) Threat weapons dynamics and engagement including basic C3 characteristics; (3) Electronic warfare (EW) defensive system processing and environment interaction; (4) Countermeasures effectiveness calculations; and (5) EW systems audio and video interface to the aircrew. The level of fidelity for this simulation is sufficient to accommodate mission rehearsal for qualified aircrews in addition to programmed, repeatable training to qualify or upgrade new aircrew members.

The IECSS real-time software for one WST is hosted on a single VME chassis with multiple 68030 CPUs, and these general purpose processors communicate with the simulation host computer through shared memory. The IECSS has been developed using a building-block approach which separates threat modeling. In this way, enhancements can be made to any model without significant impact to other existing modes. The software suite also includes off-line editors and diagnostic tools in addition to the real-time functions. Off-line threat setup involves populating a file structure which contains threat laydown and characterization data. New threats are easily added to the database through menu-driven editors.

An Inter-Simulation Network (ISN) connects up to eight IECSS-equipped trainers through a fiber-optic based reflective memory technique. All eight players share a common electromagnetic simulation but individually process the environment based upon position, occulting, and defensive models assigned. This enables the WSTs/ATDs to mission rehearse or fly in formation through a consistent environmental laydown.

ABOUT THE AUTHORS

David Galloway is currently a lead hardware engineer at TRW. He was the Systems Engineer whose responsibilities included technical management of the software and hardware team which designed and implemented the IECSS. As a Research Engineer at Georgia Tech, Mr. Galloway's responsibilities included designing and implementing real-time digital signal processing, data acquisition/control hardware and software for a phased array receiver. Mr. Galloway received a M.S.E.E. in 1988 and a B.S.E.E. in 1986 from Auburn University.

Pat Heffernan is currently a project manager at TRW. He was the lead engineer for the IECSS threat simulation. Mr. Heffernan was a USAF Electronic Warfare Officer for six years. He flew operationally with the 8th Special Operations Squadron (MC-130E). Mr. Heffernan received a B.S. in 1985 from the US Air Force Academy and another B.S. in 1990 from the University of West Florida.

Al Nuss is currently a project manager at TRW. He was the lead engineer for the IECSS EW systems simulations. Mr. Nuss was an EW engineer for the USAF and was responsible for all aspects of the ALQ-187, ALQ-101 and QRC 80-01 electronic countermeasures systems. Mr. Nuss received his B.S. in Aerospace Engineering from Purdue University and an M.S. in Management from Troy State University.

Chad Summers is currently a project engineer at TRW. He was the lead engineer for the IECSS off-line support system (database management system). Mr. Summers was a software communications engineer responsible for digital communications of LAN devices. He received a B.S.E.E. in 1986 from Auburn University.

ELECTRONIC COMBAT SIMULATION IN A NETWORKED, FULL MISSION REHEARSAL, MULTI-SIMULATOR ENVIRONMENT

David W. Galloway, Patrick G. Heffernan, E. Allen Nuss, Charles M. Summers
TRW Avionics & Surveillance Group, Warner Robins Avionics Laboratory
Warner Robins, Georgia

INTRODUCTION AND SYSTEM OVERVIEW

The Integrated Electronic Combat Simulation System (IECSS) provides a full electronic combat (EC) simulation capability for the MH-53J Pave Low and MH-60G Pave Hawk Weapon System Trainers (WSTs). A follow-on effort currently under development will install the same capability on the HC-130P Aircrew Training Device (ATD). All three of the aircrew training simulators are installed at the USAF 542nd Crew Training Wing, Kirtland Air Force Base, Albuquerque, New Mexico.

The basic system requirement for IECSS was to provide the government with the capability of realistically simulating a full EC scenario in real-time using a cockpit mockup as the user interface for the aircrew. An extension to this basic requirement was to provide the capability of supporting multiple WSTs "playing" in the same gaming scenario. In other words, the threat simulation software would have the capability of encountering any one of up to eight WSTs/ATDs.

The basic architecture of the WSTs/ATDs, shown in Figure 1, implements the common EC environment as a shared memory data structure

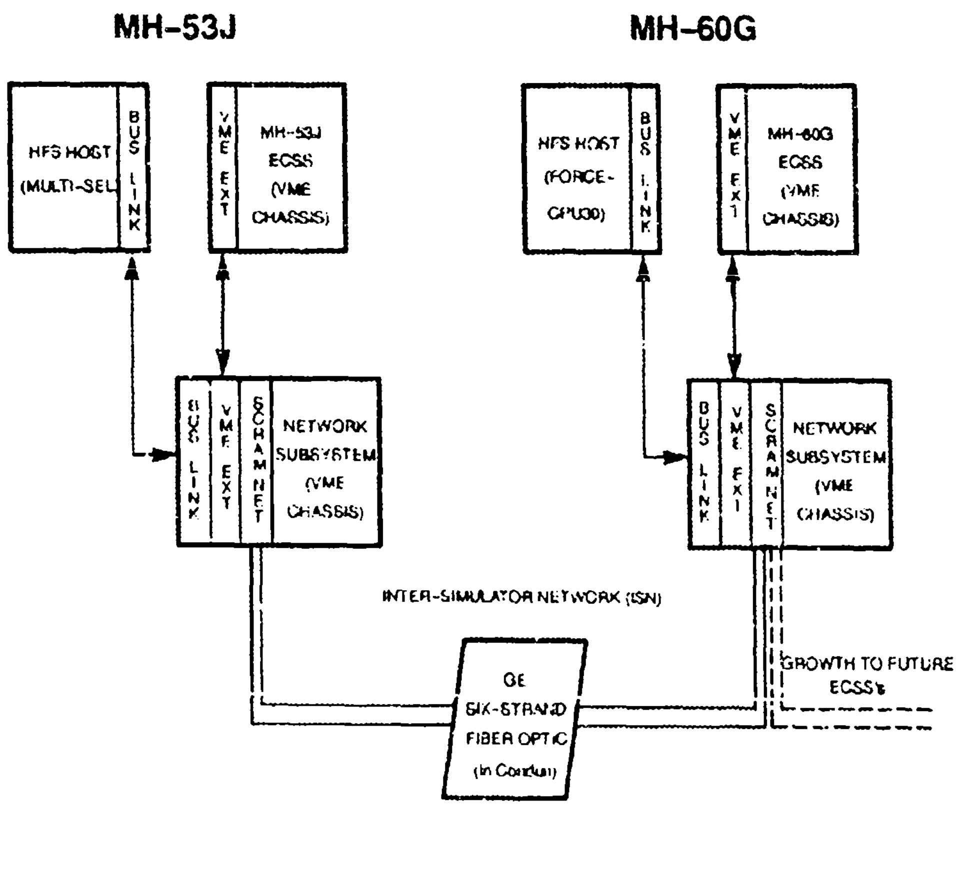


Figure 1 Weapon System Architecture

between WSTs/ATDs via the Inter-Simulator Network (ISN); a fiber optic based reflective memory. It should be noted that the IECSS nomenclature implies the simulation of multiple, integrated Electronic Combat Simulation Systems (ECSS). In a multiship, networked simulation, one of the ECSSs is declared as master of the entire threat simulation environment. It is the master's responsibility to assimilate all scenario information from the slave ECSSs to derive an active threat list. During the generation of the active threat list, the master is also responsible for determining targeting assignments against all active aircraft on the network.

Each ECSS also provides its respective WST/ATD with the ability of simulating all of the electronic warfare (EW) avionics installed on the aircraft in addition to providing a local threat simulation capability. The threat simulation will support a scenario of 400 unique threats with a maximum of 64 active at any given instant in time. The simulation also provides a closed looped simulation of countermeasures effectiveness using the best available effectiveness data.

The ECSS software architecture, shown in Figure 2, is based on the concept of simulating the EC environment using a shared memory data structure modeled as the electromagnetic environment. This data structure, referred to as the Common Environment Data (CED), contains the information pertinent to describing the battlefield environment to the level of fidelity required for a training simulator. The CED includes structures for storing active threats, active emitters, active weapons, and any active countermeasures (CM) being applied to the scenario. The CM data structures provided include RFCM, IRCM, and chaff and flare expendables.

To fully support the expansion and flexibility of the IECSS design, a full suite of off-line

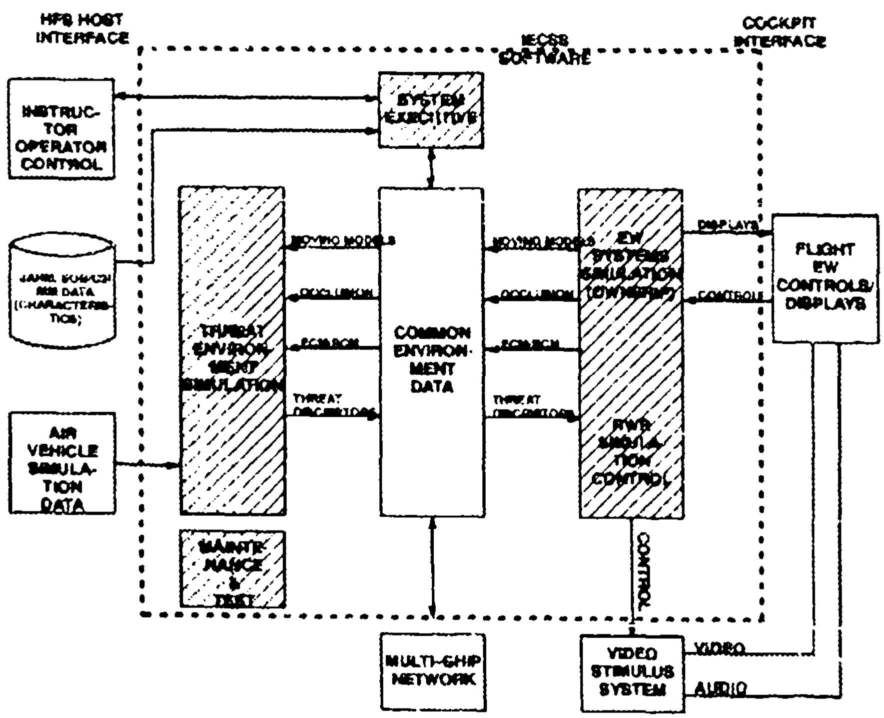


Figure 2 ECSS Software Architecture

database support tools were developed to provide the end-user with the capability of maintaining the databases required for IECSS. These tools provide the user with the capability of generating new and updating existing threat simulation parametrics. It also provides the ability for generating a scenario mission load. Each mission load contains up to 25 different mission scenarios or laydowns. Each laydown defines the location of a maximum of 400 unique threats and defines the desired Command, Control, and Communications (C3) network for each scenario.

THREAT SIMULATION

The threat simulation models activate and engage the WSTs/ATDs in a controlled manner in order to simulate actual battlefield threat environments. Although there are many threat simulation models available, the unique nature of networked aircraft operating over large areas and distances compounds the threat simulation problem because each WST/ATD must contend with its own subset of threats. In addition, the threat simulation model must be independent of other EW simulation functions so that it can be used for any WST/ATD. The IECSS threat simulation model is one approach to solving the networked problem while providing a generic model.

The threat simulation models begin with a mission load which is built off-line by the user. The mission load contains information such as threat scenario laydowns, threat parametrics, weapon parametrics, C3 information, and other EW-related data. The mission load is read into real-time data structures during initialization. An important real-time data structure is the threat laydown. The threat laydown data structure contains threat identification and positional data. This data structure is available to every WST/ATD via the ISN, thus, the threat laydown serves as the starting point for all real-time threat functions.

When the network is being operated in the master-slave role, the master WST/ATD is responsible for generating a master threat environment from the slaves' local threat environments, executing tactics algorithms for the threats in the master threat environment, generating airborne interceptor flight paths, and managing certain commands from the instructor operator station (IOS). The slave WSTs/ATDs are responsible for generating their own local threat environments, producing weapon flyouts, and reacting to certain inputs from the IOS. When the WST/ATD is operating in the standalone mode, then all functions are accomplished by the standalone ECSS.

Figure 3 describes the general approach for production of the master electronic warfare

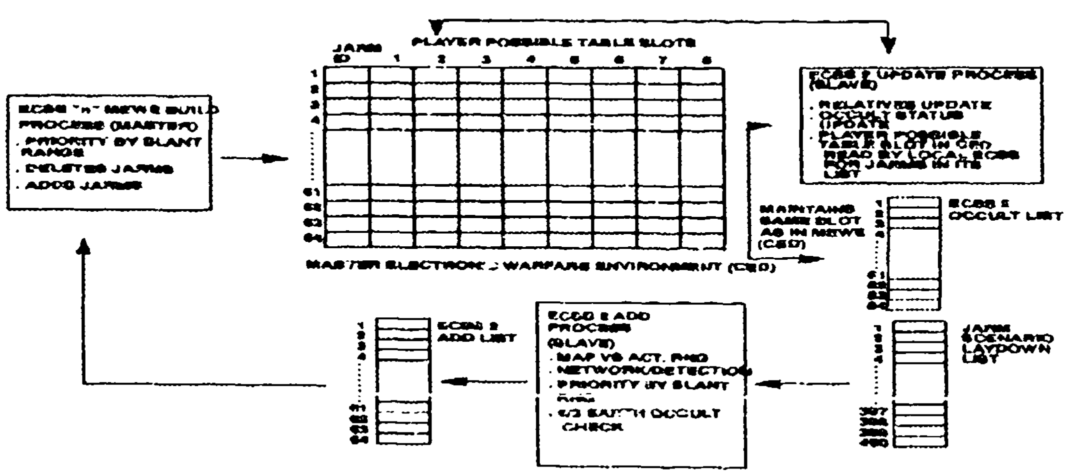


Figure 3 Master Electronic Warfare Environment

(threat) environment (MEWE). Each slave WST/ATD produces a local threat environment consisting of up to 64 threats from a maximum threat laydown of 400 threats during the add process. It was necessary to reduce the number of potential threats to a manageable limit to meet real-time limitations in processing. The reduction of potential threats from the laydown to the local threat environment (add list) is accomplished by comparing map ranges from the WST/ATD to each threat in the laydown to a threat activation range, checking initial occulting (terrain masking) information, and exercising initial C3 data. In addition, threats already in the MEWE are ignored by the add process. The resulting add list is prioritized by slant range, from the threat to the WST/ATD, and the excess threats over 64 are eliminated from further consideration.

The add lists are obtained by the master over the ISN. The master takes each add list and prioritizes each threat in the add lists by slant range. Only the 64 highest priority threats are slotted into the MEWE, which is part of the overall CED structure. The master also deletes threats from the MEWE when they are occulted from their contributing WST/ATD for a predetermined time period or when the WST/ATD leaves the threat's activation range.

After the MEWE is constructed, the master is responsible for target assignment. A WST/ATD is listed as a player possible when a threat affects it. From this player possible list, the master determines, based upon slant range, which WST/ATD will be targeted by a specific threat. Threats continue to target a WST/ATD until it is shot down, occulted, or leaves the activation range of the threat.

After target assignment, the targeted WST/ATD ECSS calculates the relative positional information (relative azimuths, relative elevations, etc.) for the threat and requests occulting information from the Digital Radar Land Mass Simulator (DRLMS) during the update process. This information is then transmitted back to the MEWE for use by the master threat simulation. The process is then repeated at a resolution required to support the overall system fidelity.

Each threat is built off-line using a series of tactics algorithms which dictate how the threat operates in the simulation. The master executes these algorithms for each threat in the MEWE depending upon positional factors, ECM/ECCM factors, and probability of detection. Atmospheric factors, such as range attenuation, rain, visibility, and terrain clutter affect the probability of detection. The master also determines when a threat fires/launches at the targeted aircraft. The goal was to make this part of the threat simulation as representative of the actual threat system operation, so that there was little to no perceivable difference to the aircrew between the actual and simulated threat environments.

An essential part of the threat simulation is the C3 system. The threats can be defined in the scenario as either autonomous or networked. An autonomous threat operates by itself based upon its own activation range and parameters. Networked threats rely upon other threats in the network, called controllers, to activate them. The controllers are defined in the mission load. Networked threats rely on the controllers to detect the incoming targets and activate them. Activation, engagement, and weapon firing are dependent upon the level of conflict selected for a threat scenario. The instructor can define the level of conflict off-line and then modify it during the mission rehearsal session through the IOS.

The master also generates the airborne interceptor (AI) flight paths. During the mission load build, AIs can be defined for the simulation. These AIs, during real-time, automatically activate and engage a targeted WST/ATD. The flight path generation routines are six DOF models and the information they provide is not only used in the threat simulation, but transferred to the Computer Image Generator (CIG) so that the AI can be presented visually in the simulation.

Each WST/ATD ECSS calculates, as stated above, relative positional information for a threat that could affect it (player possible). The relatives consist of aircraft azimuth and elevation relative to the WST/ATD X,Y,Z coordinate system, relative azimuth and elevation relative to true north (using a Cartesian coordinate system), map (X,Y) and slant (X,Y,Z) ranges, aircraft radar cross section (RCS) and infrared (IR) signature, and threat radar/IR field of view checks and power densities at the WST/ATD. The relatives are used throughout the threat and EW equipment simulations.

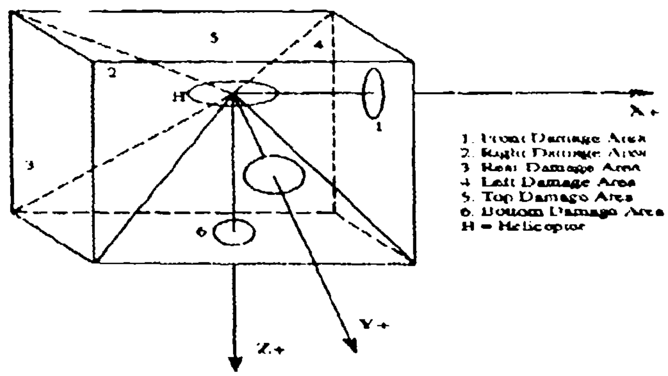


Figure 4 Damage Area Scheme

Each WST/ATD ECSS also calculates the weapon flyout trajectories and damage assessment for a weapon that has been fired/launched at it. The flyout trajectories are six degree of freedom (DOF) trajectories and are used by the CIG to present the flyout visually. There are five trajectones simulated. The pursuit trajectory determines the target line-of-sight from the weapon and directs the weapon velocity vector toward the target. The three-point trajectory determines the target's line-of-sight from the weapon, computes the weapon's time-to-impact from the line-of-sight, calculates the target velocity and the weapon speed, and directs the weapon velocity vector toward the predicted impact point. The proportional navigation trajectory determines the target line-of-sight from the weapon, computes the line-of-sight angle rate of change and vector direction of change from the line-of-sight, tracks weapon velocity history data, limits and filters the line-of-sight angle rate of change, and modifies the weapon's velocity vector direction from the filtered line-of-sight angle rate of change and vector direction of change. The beam rider trajectory determines the target line-of-sight from the weapon's control point and modifies the weapon's velocity vector direction to remain along this line-of-sight. The ballistic trajectory determines the weapon's velocity vector in a straight line path.

The damage assessment model for a WST/ATD initially begins at the termination of a weapon flyout. A Monte Carlo technique is used to determine if the weapon makes a direct impact upon the target. Direct impacts result in direct kills. If the weapon missed but still detonated, a damage assessment model is used to determine the amount of damage the target took. This model breaks the target up into six areas that surround the target. The damage area is determined at weapon termination. The amount of fragmentation/blast damage is calculated for the damage area and scaled. Each damage area has a set of critical equipment that is susceptible to damage and rated as to how much scaled fragmentation/blast it would take to "kill" that component. The amount of scaled fragmentation/blast damage is used to determine which components are "killed" or not "killed" and those that are "killed" are automatically malfunctioned. The aircrew member must then respond to the affects of a damaged aircraft which could ultimately become a "kill". Figure 4 illustrates the damage area concept for the damage assessment model.

EW DEFENSIVE SYSTEMS AND COUNTERMEASURES SIMULATION

The design approach used for the EW defensive systems models decoupled the threat models from the operation of the defensive models. Early EW simulations routinely structured their threat models to drive specific reactions in the defensive systems models. Parameters selected for threat simulations were specifically tailored to drive a required radar warning receiver display and audio response, or dictate a specified countermeasures effectiveness. As part of the threat structure, for a given threat operating mode and countermeasures systems technique, selection of a single exact

effectiveness factor would be applied to the threat for the entire engagement. It must be emphasized that these effectivity factors were placed in the threat structure and not part of the defensive systems model. The defensive systems models, in large part, only processed switch changes and lamp displays.

To accommodate this decoupling, the design for radar warning receiver (RWR) models must begin with an in- depth analysis of the operational flight program (OFP). For each module of the OFP, a trade analysis is required to determine whether: (1) the actual OFP code can be used line by line; (2) an emulation of the module be created; (3) the module be functionally simulated; or (4) the particular module may not be required for simulation/training. Of course, the simulation host language and ease of translation to the host language plays a critical role in use of the OFP line for line. Programmability of modern RWRs is a critical factor in the rehost/emulation analysis. The approach used in the IECSS was driven by the design decision to incorporate user update capability to the emitter identification data (EID). When a change is made to the aircraft's EID data, the same change can be loaded in the simulation's EID since the same data structure has been maintained. The modules of the OFP that directly access the EID are therefore rehosted to the simulation's host language. This design approach creates a simulation near emulation of the aircraft's RWR. As a result, the RWR simulation operates independently of the threat environment. The RWR simulation samples the threat environment, reads all threat parametric data, and from this point on processes the data, displays the threat symbology and creates the associated threat audio without further direct interaction with the threat environment until a new processing cycle is initiated.

For active countermeasures systems models, all switchology and display simulation, threat processing, and countermeasures assignment are accomplished internal to the model. Data structures independent and decoupled from the threat structure are used to store and maintain the effectivity factors used by the countermeasures effectiveness module of the IECSS simulation. For electronic and infrared countermeasures systems, the technique type and parametrics associated with the technique are determined by the defensive system model and transferred to the countermeasures effectiveness routines. For expendable countermeasures, the timing between releases and the quantity for each release is controlled by the defensive system model. The quantity for each dispense event is transferred to the countermeasures effectiveness routines as they occur.

The key link to our decoupled threat and system design is the countermeasures effectiveness (CME) subroutine. The CME receives inputs from the IECSS EW defensive systems and the threat related data from the software partitioned common environment data (CED). The CME scans all operating defensive systems for valid countermeasures entries in the effectiveness data tables. A list of possible countermeasures is stored for further processing. For example, if a chaff cloud is active, the active threats are tested for matching operating mode and stored in a countermeasures possible list. Once all the active threats and defensive systems countermeasures are determined, each possible countermeasure is dispatched to the appropriate type of countermeasure effectiveness subroutine (RF, IR, chaff, or flares). These subroutines compare the ownship's radar cross section and IR signature to the countermeasure's chaff and flare signatures respectively as well as the jammer-to-signal ratio and IR intensity for RF and IR jammers. The tabulated effectiveness factors are then adjusted for probability of success based upon real world flight and simulation test data. The ownship's RCS and IR signature is updated every effectiveness call based upon current threat/ownship engagement geometry. The CME module is executed at a one hertz rate, and calculated threat aim point adjusted every cycle based upon the current geometry and environmental conditions.

DATABASE SUPPORT

The purpose of the IECSS Database Support System is to provide an off-line editing capability of the information needed to support a real-time weapon system simulation. The data is structured using relational database techniques and is available for update via menu driven interactive screen input and standard SQL commands. The IECSS Database Support System uses the ORACLE Relational Data Base Management System and ORACLE products such as SQL*FORMS and SQL*MENU to

accomplish this task. The database support system is divided into two logical categories; Library Maintenance and Mission Preparation.

Library Maintenance tasks involve the editing of data which is used to support the following simulation capabilities:

1) Threat Parametrics
2) Site and Platform Maintenance
3) Aircraft EW Configuration
4) Electronic Order of Battle/C3 (EOB/C3) Parameters
5) EW System Characteristics and Emitter Identification Data

Threat Parametrics are used by the real-time threat models to perform an accurate simulation of threats as they interact with the electronic warfare environment. The Site and Platform library is essentially an association of one or more threats with a Site or Platform name. The Sites and Platforms built can be positioned in a mission scenario causing all associated threats to be incorporated. Aircraft EW Configuration parameters define specific system setup data for a particular aircraft. This includes the positioning of countermeasure dispense hardware onboard the aircraft. The EOB/C3 library defines engagement modifiers and communication delay times based on a particular conflict level. The EW System characteristics and Emitter Identification library contains the data used by the various EW system models to perform an accurate simulation of the system against threats in the environment. The data maintained in each of the Library Maintenance libraries is read during IECSS initialization at the beginning of a mission training or rehearsal session.

Mission Preparation involves the building of mission scenarios, or laydowns, based on Library Maintenance threat data, and the generation of these scenarios and supporting data sets into a mission load. A mission load may contain up to 25 different scenarios and includes the threat parametrics, weapon parametrics, EW system characteristics, Emitter Identification data and threat positional data required to support the simulation. The mission load files are read during IECSS initialization.

The mission scenario is the EW environment defined by the aircrew instructor to accomplish a particular training or rehearsal objective. The scenario is built by positioning threats using the latitude/longitude coordinate system. Each threat is assigned other field information such as altitude, speed, heading visual identification, communication delay times and engagement modifiers. Threats within a single mission scenario may be networked together and assigned corresponding EOB/C3 parameters.

SUMMARY/CONCLUSIONS

The IECSS supports a full EC aircrew training capability providing for training in the usage of the installed EW avionics and the proper techniques required for countering the simulated threat sites. By supporting the ability of simulating multiple aircraft in a single gaming scenario, the IECSS provides a mission rehearsal capability that is unparalleled in the simulator world.

Another major achievement in the IECSS software implementation is the modularity of the design allowing for future software upgrades. This design approach allows for new threat simulations to be added to the environment via the off-line editors while the real-time design allows for insertion or deletion of EW avionics models based on changes to aircraft configuration. Also, by keeping in mind the desire to add new SOF aircraft to the ISN, the design also supports adding new aircraft configurations to the software development environment. Finally, by having a single data area (CED) for all EC environment information, the IECSS design can be easily integrated with other dissimilar simulations/simulators by using the Distributed Interactive Simulation (DIS) standards and protocols.

A NEURAL-NETWORK-PROGRAMMABLE PROCESSOR FOR REAL-TIME CORRELATED SENSOR SIMULATION

Budimir Zvolanek
Erwin W. Baumann
McDonnell Douglas Training Systems
McDonnell Douglas Corporation
St. Louis, Missouri

ABSTRACT

Simulation of Infra-Red (IR), Synthetic Aperture Radar (SAR), and Out-The-Window (OTW) visual imagery plays an important role in the planning and rehearsal of missions and personnel training. The challenge is to develop database and image generation systems that extremely rapidly and in real time process geo-specific Multi- Spectral Imagery (MSI) over large areas into simulated sensor imagery to achieve high real-world accuracy and sensor / OTW-visual correlation. To meet this challenge, a novel architecture called a Neural Network Look-Up Table (NNLUT) which implements spectral conversion by neural networks has been developed. The NNLUT processor is described and examples of highly correlated IR and SAR imagery simulated in real time from MSI by the NNLUT are demonstrated.

ABOUT THE AUTHORS

Budimir Zvolanek is a Technical Specialist, Technology Development and Integration, at McDonnell Douglas Training Systems (MDTS). Mr. Zvolanek has concentrated his career on the application and development of electronic imaging systems, image exploitation, sensor signal processing, and simulation databases. He has been a Principal Investigator in image simulation and database R&D, a program manager for DoD Standard Simulator Data Base Project 2851 contracts at MDTS, and is a co-inventor of the NNLUT system. Mr. Zvolanek received his B.S.E.E. and M.S.E.E. degrees from Washington University, St Louis, Missouri.

Erv Baumann is a Senior Technical Specialist, Technology Development & Integration, at MDTS. Mr. Baumann has specialized in the application of artificial intelligence to sensor simulation, threat modeling, machine vision, automated planning, and is a co-inventor of the NNLUT. Mr. Baumann received his B.S.E.E. degree from the University of Minnesota.

A NEURAL-NETWORK-PROGRAMMABLE PROCESSOR FOR REAL-TIME CORRELATED SENSOR SIMULATION

Budimir Zvolanek
Erwin W. Baumann
McDonnell Douglas Training Systems
McDonnell Douglas Corporation
St. Louis, Missour

INTRODUCTION

The simulation of sensor and Out-The-Window (OTW) visual imagery plays an important role in the planning and rehearsal of missions. Typical imaging sensors include Forward-Looking Infrared (FLIR), Synthetic Aperture Radar (SAR), night-vision image intensifier, and television camera electro-optical sub-systems.

Mission and User Needs

In order to maximize the probability of a successful mission, mission personnel need to observe simulated imagery to become familiar with how the world will appear through on-board sensors as well as out the windows of their platforms before they proceed with the actual mission. This "image familiarization" procedure may be performed while planning the mission and evaluating, or rehearsing, it.

Because the simulated imagery needs to portray the real world in terms of time as well as geographic accuracy, only the most recently acquired imagery of geographic areas in the world should be used as the basis for the simulation (Figure 1). The acquired Multi-Spectral Imagery (MSI) data is pre-processed to compensate for sensor peculiarities and geo-positioned over stored three-dimensional (3-D) terrain elevation and model height data. The combined imagery and 3-D data is reformatted into a run-time Image Generator (IG) database and is then available for mission planning and rehearsal.

The challenge is to develop systems that perform such pre-processing, geo-positioning, and run-time IG database generation extremely rapidly so that the simulated sensor and OTW-visual imagery is made quickly available to mission users before it becomes outdated. The information extracted by the personnel from the simulated imagery also needs to correlate well among the sensors and OTW-visual domains. Furthermore, such processing should be easily performed by mission personnel deployed in the field - without requiring a high level of personnel expertise in simulation database modeling. And finally, the correlated sensor and OTW-visual simulation and database system should be designed to allow trade-offs between cost, fidelity, and other simulation performance measures.

Unique Sensor Simulation Approach

This paper describes a neural-network-programmable processor called NNLUT -a Neural-Network Look-Up Table - to satisfy such multi-sensor simulation needs. The NNLUT processor hardware and software have been uniquely implemented for accurate and real-time simulation of geo-specific, correlated, Infra-Red (IR) and SAR sensor imagery directly from multi-band, Multi-Spectral Imagery (MSI).

The NNLUT system approach achieves the rapid availability, high correlation, and high geo-specific accuracy of the simulated multi-sensor and OTW-visual imagery by directly processing the source MSI output by the CIG in real-time, thereby minimizing effort-intensive, non-real-time, database modeling activities (Figure 2).

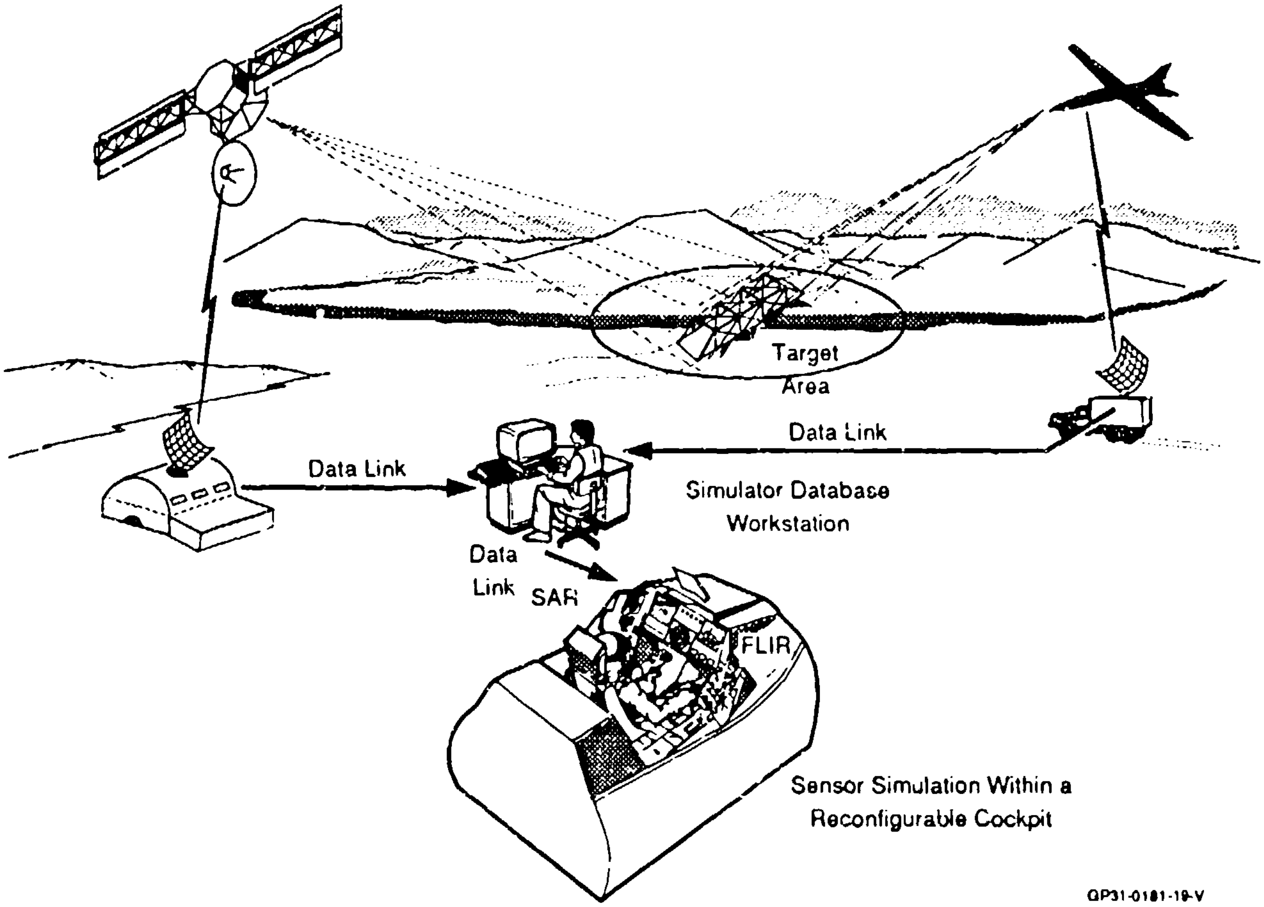


Figure 1. Field-Deployable Image Simulation Facility

The rapid simulation availability is achieved in two ways. First, generating only a single run-time IG database, rather than separate multiple sensor and OTW-visual IG databases, makes the "common" database available for real-time, direct, processing by the NNLUT.

The second way, in which rapid availability is achieved, is by directly processing geo-specific MSI over large gaming areas (backgrounds) while features and models are inserted only in small, high-interest areas. Run-time database generation steps, represented by the "Reform ing" block in Figure 2b, include geo-positioning of the input MSI over terrain elevation data using accurately known ground-control points, ortho-rectification to compensate for MSI sensor oblique view angles, mosaicking, and contrast-balancing of multiple MSI scenes into a gridded gaming area database, partitioning the MSI/elevation database into IG texture blocks, and performing IG polygon-to- texture mapping assignments.

Geospecific accuracy is achieved by directly processing the MSI into sensor image intensities thereby preserving information relating to surface materials. This approach is in contrast to traditional sensor database generation which reduces MSI into sets of discrete material codes, attributes, and generic textures, thereby discarding valuable surface-descriptive information. This paper therefore focuses on optimal, real-time utilization of MSI for sensor image generation.

The next section describes the application of neural networks to spectral conversion based multiple-sensor image simulation. An equivalent look-up table for implementation of the neural network processing is then described, leading into a description of the real-time processing NNLUT architecture and its hardware implementation. Simulations of correlated overhead IR and SAR images as well as FLIR generated in real time by the NNLUT system are presented. Issues regarding cost vs image correlation, accuracy, and fidelity as well as NNLUT system implementation are discussed. Finally, plans for future enhancements and applications of the NNLUT system to rapid database generation and other imaging tasks are presented.

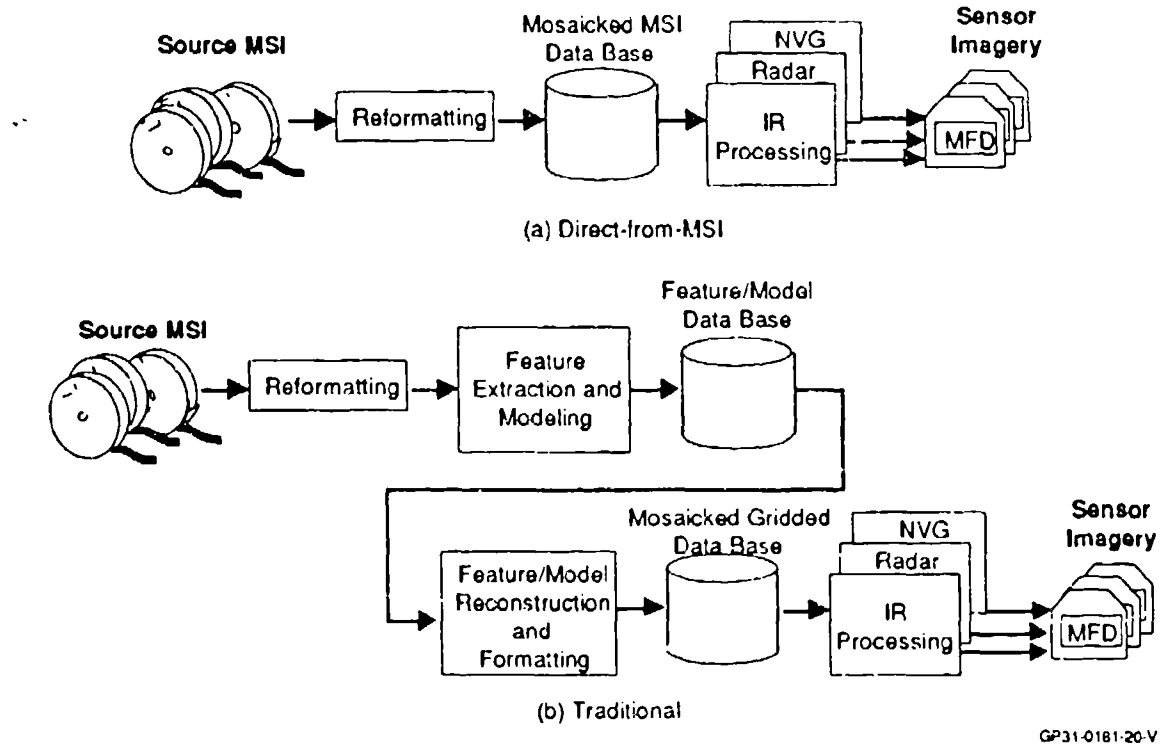


Figure 2. Direct-from-MSI vs Traditional Multi-Sensor Image Simulation

DIRECT-FROM-MSI MULTI-SENSOR IMAGE SIMULATION

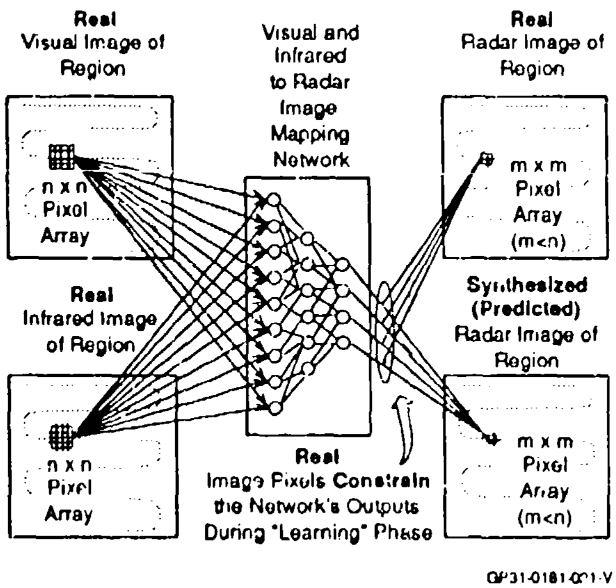


Figure 3. Neural Network Training for Direct-from-Imagery Sensor Simulation

To realize direct-from-MSI multi-sensor image simulation, the NNLUT system implements image-to-image mapping transformations, sometimes called "spectral conversion", using neural networks.

Spectral Conversion

The purpose of spectral conversion processing is to convert source MSI input intensities into the desired sensor output intensities. An MSI pixel consists of image intensities from two or more spectral bands. For example, commercially available MSI sensors produce imagery in the visible 0.5-0.7, near-IR 1.0, and a short-IR 2.4 micron wavelength bands. The desired simulated sensor pixel intensities may be in the long-wave IR 8-12 micron or 3-cm X-band radar wavelength bands.

Neural Networks

One approach to spectral conversion is to train and apply computational structures called Neural Networks (NNs) as associative memories to perform the MSI-to-IR and MSI-to-SAR inversion[1]. Figure 3 illustrates NN training to transform visible and IR-band input image intensities into radar intensities. While training, the network is iteratively presented with exemplars of corresponding visible-

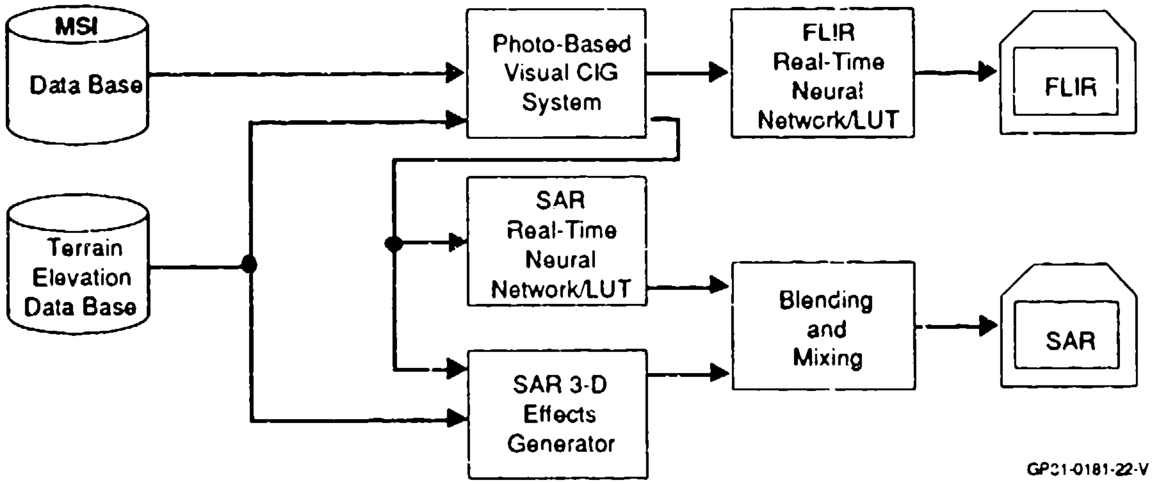


Figure 4. Correlated FLIR and SAR Image Generation from a Single MSI/Elevation Database

plus-IR input and SAR output pixels to adjust its internal weights. After the error between the predicted and actual intensities stabilizes at a minimum value, the network is ready for SAR intensities prediction when presented with similar visible-IR input combinations.

Figure 4 illustrates how the networks are used to post-process outputs from a texture-capable host Image Generator (IG) in real time. Each channel of a two-channel IG transforms the elevation and MSI texture database into an image of a scene computed from the imaging sensor's 3-D perspective. For a Forward-Looking IR (FLIR) sensor, a highly oblique terrain view is produced from one channel. The NNLUT converts the MSI pixel values output by the IG into monochromatic IR intensities. For SAR simulation, an overhead view MSI image from the second channel is post-processed by the NNLUT and the transformed image is combined with an image containing simulated SAR 3-D effects such as far-shore brightening, aspecting, and shadowing. A high degree of FLIR and SAR correlation is achieved because the same IG run-time database is used for both FLIR and SAR simulations.

To successfully implement the MSI-to-sensor transformations in real time, two key problems were solved. First, suitable NN architectures were selected and implemented. While we achieved reasonable results in SAR and near-IR simulation using a back-propagation trained multi-layer feed-forward neural network[2], we also developed a unique neural network called Stochastic Associative Memory (SAM)[3] for improved FLIR and SAR simulation. The SAM network not only improved the MSI-to-sensor intensity prediction accuracy for SAR and FLIR but also enabled the learning and prediction, in real time, of an actual sensor's image noise.

The second problem to overcome was the implementation of the MSI neural network processing at real-time video pixel rates. The next section describes how such need for real-time NN implementation was circumvented by using a large look-up table equivalent to a neural network.

NEURAL-NETWORK AND LOOK-UP-TABLE EQUIVALENCE

The underlying concept behind the NNLUT processing is the mathematical equivalence between an M-bit-input, N-bit-output neural network and an M-bit- input, N-bit-output Look-Up Table (LUT).

A very large, 2^M-entry (2^MxN-bit), LUT can contain the transfer function of any associative-memory type neural network architecture having M input bits and N output bits. For example, a 24-bit-input, 8-bit output neural network trained to transform three-channel MSI 24-bit data into 8-bit, 256-level intensity IR image can be implemented by a 24 bit-in, 8-bit-out, 16-Megabyte LUT as shown in Figure 5.

The key to using such a large LUT is to train a neural network on a limited set of exemplar MSI-input / sensor-output pixel combinations and then use the network to compute and define the LUT contents for all possible input/output pixel combinations expected during actual simulation. As shown in Figure 5, the NNLUT approach is practical for sensor image simulation: a total of 24 input LUT bits allows the transformation of three 6-bit MSI chan-

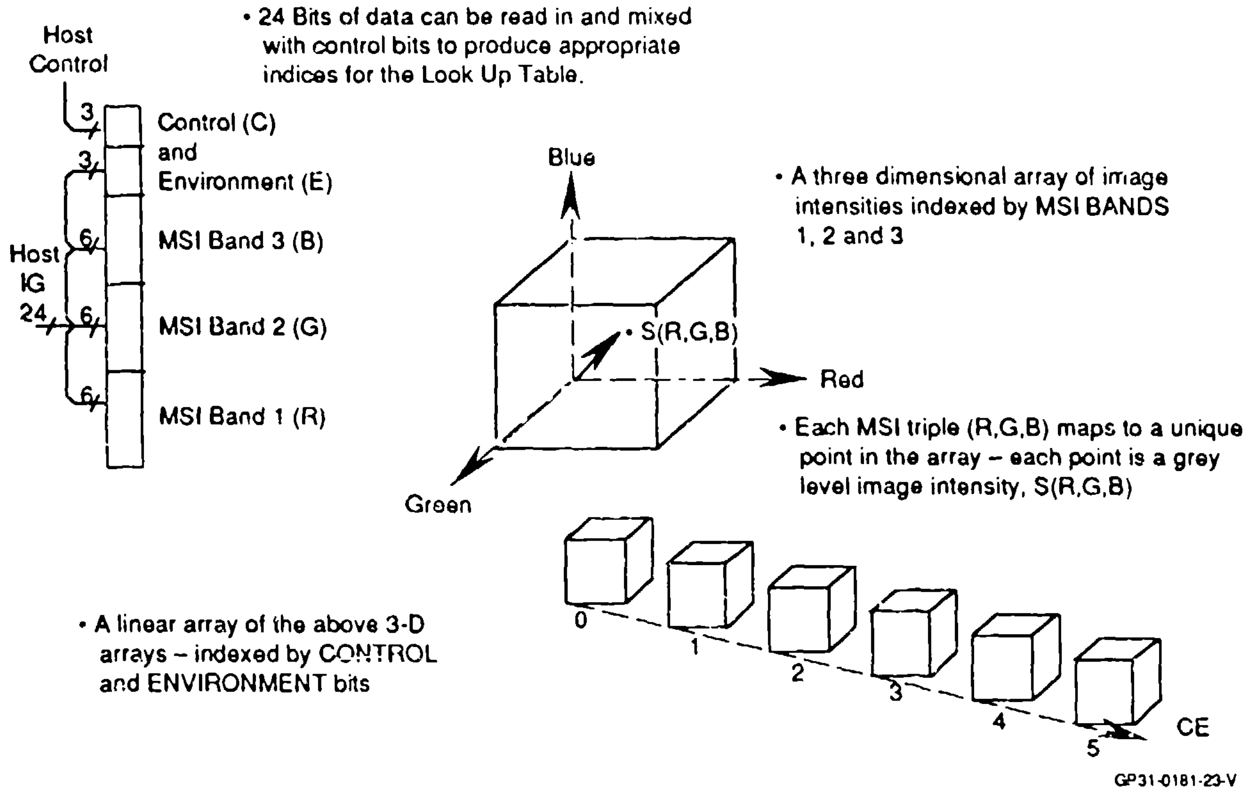


Figure 5. Mapping of MSI and Control Bits Into a Sensor Image Grey Level

nels (18 bits) plus environmental and time-of-day control bits (6 bits). Such capability plays in concert with the need to accommodate several million 12-bit LUT entries for a high-fidelity IR simulation as expressed by IR simulation experts[4,5].

Due to dramatic advances in the availability of high-speed, high-density memory modules, a compact and low-cost hardware implementation of a very large LUT to transform MSI data at real-time video rates has become feasible. Therefore, the requirement to implement a complex NN architecture in hardware executing at real-time video pixel rates for spectral conversion has been eliminated. The next section describes the real-time NNLUT hardware architecture and integration within a sensor simulation system.

REAL-TIME CORRELATED SENSOR SIMULATION SYSTEM ARCHITECTURE

The NNLUT system architecture, represented by the block diagram in Figure 4, integrates with the host IG as shown in Figure 6. To implement simultaneous IR and SAR image simulations, two parallel NNLUT systems can used. The host IG provides MSI images from two channels at independent viewing geometries for transformation by the two NNLUTs.

Architectural Features

A unique feature of the NNLUT system is its capability to process analog RGB color video signals from the host image generator, thus eliminating the need for custom digital video interfacing to a host IG. The host IG effectively encodes the MSI data into an RS-170A color video signal (Figure 6). The NNLUT board(s) are integrated with real-time color image frame grabber and display boards. Using the frame grabber, the NNLUT system digitizes host IG RS-170 RGB color video output into 24 bits. The digital data is passed to the NNLUT boards over a 24-bit digital video bus.

Another feature of the NNLUT system is the implementation of a real-time pseudo-random number generator on the NNLUT board to provide SAR real-time statistical texture and coherent noise generation learned by the SAM neural network. In addition, host-selectable LUT input bits can be set or reset, overriding the input MSI data bits, to implement environmentally-dependent simulation control (Figure 5).

For SAR 3-D special effect generation, an image processing accelerator board is included with the

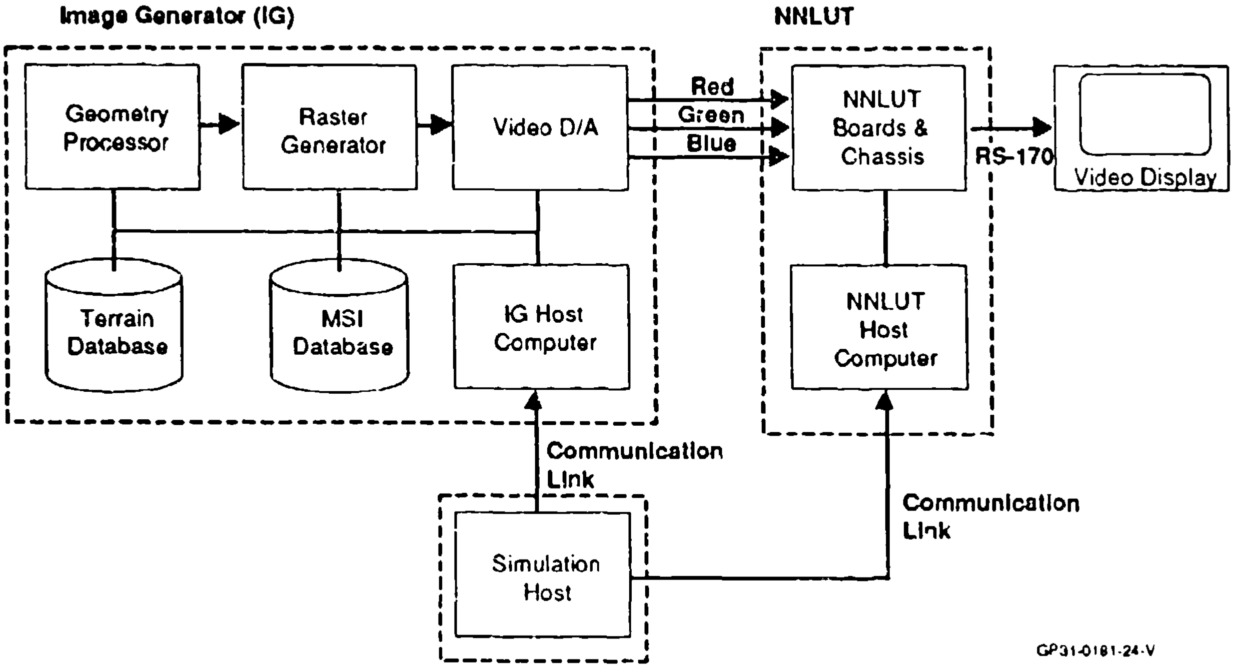


Figure 6. NNLUT System Integration With a Host IG

NNLUT in a single enclosure. Directional edge enhancement and local neighborhood processing performs far-shore brightening and aspecting effects simulation. Therefore, radar 3-D effect generation becomes possible even when high-resolution 3-D terrain elevation data is not available or is difficult to acquire.

NNLUT Implementation

The NNLUT system is currently integrated within an IBM PC/AT-compatible computer and centers around the 4-Megabyte NNLUT board shown in Figure 7. The board can process 22-bit input MSI data into 8-bit output imagery at a continuous real-time video rate of 10 Megapixels, or 30 Megabytes, per second. The 4-Megabyte LUT contents are loaded from the host PC via a 2-Megabyte PC extended memory window.

An ISA (Industry Standard Architecture) PC/AT Input/Output bus chassis allows integration of one or more NNLUT boards. Four boards can provide a complete 24-bit (16.7-million color) input MSI, 8-bit output sensor capability. The NNLUT implementation will accept plug-compatible 2-MByte memory modules for future expansion. Such doubling of the NNLUT size to 8 Megabytes per board will provide 25 MSI-plus-control bit input NNLUT system capability on four boards. The design is further expandable to accommodate 15-nsec speed memory modules for high-resolution video rates of 40 million pixels per second.

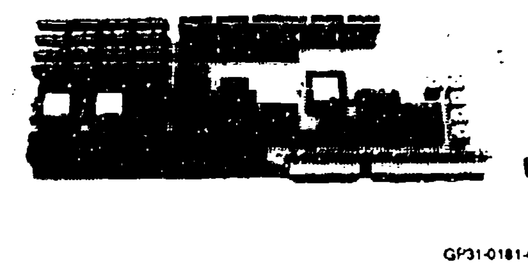
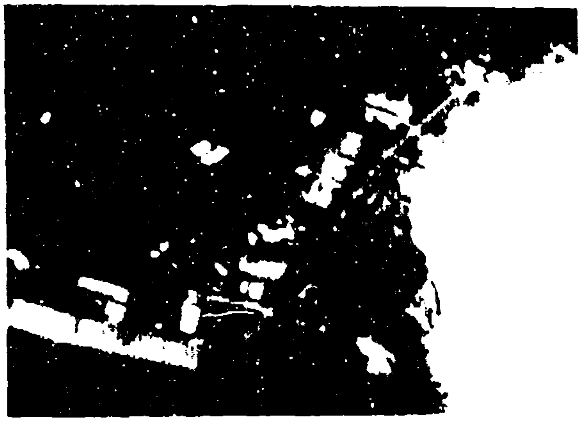


Figure 7. IBM PC/AT – Compatible NNLUT Board

NNLUT IMAGE SIMULATION RESULTS

While live demonstrations best highlight the NNLUT's real-time capabilities, Figures 8 and 9 show samples of simulated sensor imagery generated by the NNLUT system using commercially available MSI data as input.

Figure 8(a) shows an overhead-geometry multispectral image of Edwards AFB in California produced by a Flight Safety International VITAL VII texture-capable image generator. The VITAL VII was used due to its ready availability but any



(a) IG-Produced Input Multi-Spectral Image

(b) NNLUT-Simulated IR Image

(c) NNLUT - Simulated SAR Image

GP31-0181-026-V

Figure 8. Correlated Image Simulation by the NNLUT System

GP31-0181-027-V

Figure 9. FLIR Image Simulated by the NNLUT System

texture-capable IG could have been used (See Discussion/System Implementation). The 10-meter per pixel resolution false-color image (shown here in black-and-white only) is a mixture of two 30-meter resolution, 2.1-2.4 micron near-IR, Landsat TM MSI bands sharpened by a 10-meter, panchromatic, SPOT image. Figure 8(b) shows the MSI data transformed by the NNLUT system into an 8-12 micron wavelength IR sensor image. Figure 8(c) shows the same image transformed into an X-band SAR image. Note the high degree of correlation between both simulated scenes and the simulation of directional SAR aspecting effects along the leading edges of buildings. The material- dependent simulated SAR coherent speckle noise learned by the SAM neural network is also evident.

Figure 9 shows a simulated FLIR image as if produced by a narrow field-of- view IR sensor aboard a platform 2,000 ft above ground and at 5 miles range. Note the high degree of geospecific content, or accuracy, in both IR and SAR simulations due to direct processing of geospecific imagery. The building models were manually inserted into the polygonal IG database and MSI intensities were assigned to their sides to match approximate IR heat emission during early evening.

DISCUSSION

System Performance

The NNLUT approach offers a number of desirable characteristics illustrated by the implementation matrix in Figure 10.

Simulation Characteristics	Implementation Component: Single IG	Single DB	Direct-From MSI Algorithm/SW	Real-Time Hardware
High Correlation	X	X	X	
High Geo-Specific Accuracy			X	
High Temporal Accuracy		X	X	
Medium Sensor Fidelity	X			X
Rapid Simulation Availability		X	X	X
Real-Time Dynamic Effects			X	X
Compact Enclosure	X	X		X
Low Cost	X	X	X	

GP31-0181-028-V

Figure 10. NNLUT Characteristics vs Implementation

A direct consequence of the NNLUT's direct-from-MSI processing is a high degree of real-world geo-specific accuracy in simulated imagery contents. Using the same database for both IR and SAR simulation, the approach also provides a high degree of content correlation for the sensors. And because the simulated imagery can be made available extremely rapidly from the most up-to-date MSI data, the NNLUT can also provide high temporal accuracy. Furthermore, using a single IG host and database for both sensors can result in a very low cost multi-sensor simulation capability.

An issue of importance is the simulated sensor effects fidelity. For IR simulation, the NNLUT system promises dynamic, real-time simulation of diurnal and seasonal variations while IR detector and optics effects are handled by a special post-processor associated with the host image generator and NNLUT. The IR sensor fidelity is therefore potentially high.

For SAR simulation, the sensor fidelity may be estimated to be of "medium" level; 3-D aspecting and far-shore brightening effects are generated without 3-D elevation data. However, an NNLUT approximation of these effects using only MSI data may still be more useful than no simulation at all when high- resolution terrain elevation data is impossible or difficult to obtain, especially over large gaming areas.

While detailed, high-fidelity 3-D SAR effects simulation may be required for some applications, such as training personnel to use real-beam radar systems having a multitude of sensor controls, such fidelity may be an over-kill in many others. For example, in mission planning, preview, and rehearsal applications when automated digital SAR systems with few controls are used by expert systems operators, the SAR simulation provided by the NNLUT should be quite acceptable. Similarly, when large networked battle simulations containing hundreds of sensor-equipped entities are staged, the use of many very low cost but geospecifically accurate and correlated sensor simulations using the NNLUT system would be advantageous.

Looking at the matrix in Figure 10, sensor fidelity is only one of eight desirable characteristics of a total multi-sensor simulation system. While different simulation problems weight the required characteristics differently, in many cases, such as in the large networked scenarios eluded to above, the NNLUT approach can offer a better than "90-percent" solution. And when low cost is the driving factor, the single-IG / single-database NNLUT system can present a very attractive "90-percent" alternative.

System Implementation

The NNLUT system functions as a post-processor of imagery output by a texture-capable IG. Some applications require detailed 3-D effects radar simulation, but many also have a requirement for real-time, highly-correlated IR and radar simulations. Therefore, a careful analysis of the total simulation system performance vs cost trade-offs may reveal that a solution using a single, two-channel, visual host IG and two NNLUT systems is preferrable over using separate, less well correlated IR and radar image generators at higher cost.

Although the NNLUT approach appears to impose a real-time geospecific texture capability requirement on the host IG, the NNLUT could be trained on outputs of any IG with generic or no texture capability. The only requirement is that the

IG be capable of displaying 24 bit color (polygon or texture) derived from MSI data by the database generation system. In fact, this may be a very low-cost alternative for correlated sensor simulation. While under-utilizing the direct-from-MSI sensor simulation capabilities of the NNLUT, the NNLUT could still provide dynamic, real-time diurnal and seasonal IR variations and correlated SAR textures while paving the way for capability improvement when a texture-capable host IG becomes available.

CONCLUSIONS

The development and demonstration of the NNLUT architecture have shown that it is possible and practical to simulate geospecifically accurate and highly correlated IR and SAR imagery directly from MSI data in real time. In fact, both sensor and OTW-visual simulations may be simultaneously possible using only a single IG and three NNLUTs to process a single, common, geospecific MSI plus elevation run-time database. Such a low-cost capability would indeed represent a major advance in image simulation technology.

Potential future enhancements to the existing NNLUT system could include the addition of a real-beam radar scanning display capability, true 3-D radar effects computation from 3-D elevation data, and simulation of dynamic diurnal and environmental sensor effects.

While the NNLUT technology has been developed with mission planning, preview, and rehearsal in mind, the technology may have many uses in civilian applications such as medical imaging, environmental status assessment, and rapid processing of remotely-sensed imagery. One example may be automatic image segmentation and fusion for display of data from magnetic resource images, x-ray machines, ultrasound scanners, and tomography machines. Another may be a special conversion of airborne MSI into SAR imagery for comparing differences in terrain due to flooding or huricane damage. Finally, the NNLUT system has already been successfully applied to real-time, automatic, surface material classification for radar generator database updating using commercially available MSI data.

ACKNOWLEDGMENTS

The NNLUT development has been supported by McDonnell Douglas Training Systems Independent Research and Development funds.

The authors would like thank Gene Brown, McDonnell Douglas Senior Fellow, for providing the funding and Dave Coblitz for supporting the NNLUT product development activities.

Further appreciation is due to the following people who made the NNLUT development a success: Anne Rahubka, hardware; Brandt Dargue, control software; Dave Williams, NN software; Terry Schmidt, fabrication and testing at the Advanced Avionics Integration Center; and Doug Dillard, system demonstrations.

Finally, the authors would also like to thank Bill Humphry, Drew Lindbergh, and John Hillerman at Flight Safety International for providing the VITAL VII image generator equipment and run-time database generation support.

REFERENCES

1. Baumann, E. W., and Zvolanek, B., "Sensor Image Simulation Using Neural Networks", IMAGE VI Conference, Scottsdale, Arizona, 14-17 July 1992.

2. Zvolanek, B., and Baumann, E. W., "Rapid-Response Imaging Sensor Simulation", 13th Interservice / Industry Training Systems Conference, Orlando, Florida, 2-4 December 1991.

3. Baumann, E. W., and Williams, D. L., "Stochastic Associative Memory", SPIE Conference on Science of Artificial Neural Networks II, No. 1966, Orlando, Florida, 12-16 April 1993.

4. Crane, P. M., "Forward-Looking Infrared Simulation Fidelity in Aircrew Training Devices", 13th Interservice / Industry Training Systems Conference, Orlando, Florida, 2-4 December 1991.

5. Peters, D. L., "FLIR - A Different World", IMAGE V Conference, Phoenix, Arizona, 19-22 June 1990.

RESEARCH IN THE USE OF VIRTUAL ENVIRONMENT TECHNOLOGY TO TRAIN DISMOUNTED SOLDIERS

Bruce W. Knerr, Stephen L. Goldberg, Donald R. Lampton, Bob G. Witmer, & James P. Bliss
U. S. Army Research Institute STRICOM, Orlando Field Unit
Orlando, FL

J. Michael Moshell & Brian S. Blau
University of Central Florida Institute for Simulation and Training
Orlando, FL

ABSTRACT

The Army has made a substantial commitment to the use of a simulated, electronic battlefield for combat training. Current and next-generation training systems can provide a realistic combat simulation for soldiers fighting from vehicles, but not for individual dismounted soldiers. Virtual Environment (VE) technology has the potential to provide that capability. The Army Research Institute, with contract support from the University of Central Florida Institute for Simulation and Training, has initiated a research program to investigate the use of VE for training dismounted soldiers. Issues we are investigating include: are some types of visual displays and controls better suited for training or task performance than others; does visual immersion in a simulated environment improve learning of the configuration, locations of objects, and routes through that environment; what scene details are most important for the acquisition of spatial knowledge and the interpretation of terrain information; does immersion in a virtual world cause disorienting side-effects, and if so, how can they be reduced. This paper describes the initial results of our research program. We developed: a set of tasks, the Virtual Environment Performance Assessment Battery, and a questionnaire to measure "Presence", the extent to which the participant felt immersed in the VE experience. We also included existing questionnaires to measure the frequency and severity of simulator sickness. The tasks measure the underlying skills needed to move, employ weapons, and communicate in a virtual environment, but do not require previous military training. They include the perception of form, color, and distance; control of simulated movement; tracking of targets; manipulation of objects; and reaction time. Thirty participants in two experiments performed the tasks using either a spaceball or joystick. Results indicate that performance on the battery tasks is sensitive to differences between the control devices and amount of practice. The presence scale possesses high internal consistency and is sensitive to the type of virtual environments experienced. Most participants experienced some symptoms of simulator sickness. Future research plans are discussed.

ABOUT THE AUTHORS

Bruce Knerr, Stephen Goldberg, Donald Lampton, Bob Witmer, and James Bliss are employed at the U.S. Army Research Institute STRICOM, Orlando Field Unit in Orlando, Florida. Dr. Knerr is a Research Psychologist and Leader of the Simulated Training Environments Team. Dr. Goldberg is a Research Psychologist and Chief of the Field Unit. Mr. Lampton and Dr. Witmer are Research Psychologists currently involved in advancing dismounted training applications of Distributed Interactive Simulation and Virtual Environments. Mr. Bliss is a Consortium Research Fellow and also a doctoral candidate in human factors psychology at the University of Central Florida. Dr. Michael Moshell is an Associate Professor of Computer Science at the University of Central Florida and creator of the Visual Systems Laboratory at the UCF Institute for Simulation and Training. Mr. Brian Blau is a Visual Systems Scientist at the Institute for Simulation and Training.

RESEARCH IN THE USE OF VIRTUAL ENVIRONMENT TECHNOLOGY TO TRAIN DISMOUNTED SOLDIERS

Bruce W. Knerr, Stephen L. Goldberg, Donald R. Lampton, Bob G. Witmer, & James P. Bliss
U. S. Army Research Institute STRICOM, Orlando Field Unit
Orlando, FL

J. Michael Moshell & Brian S. Blau
University of Central Florida Institute for Simulation and Training
Orlando, FL

INTRODUCTION

The Army has made a substantial commitment to the use of distributed interactive simulation (DIS) for combat training, concept development, and test and evaluation. The emphasis in the initial version of DIS (SIMNET) and in the next generation Close Combat Tactical Trainer (CCTT) has been on the simulation of combat for soldiers fighting from vehicles, not for soldiers fighting on foot. Representations of dismounted soldiers in these simulations are controlled by individuals at computer workstations, supplemented by "intelligent" software. No matter how well these forces may "populate the battlefield" and prepare mounted soldiers, they provide little or no training for the dismounted soldiers themselves. In contrast, the Army expects that changes in its mission will result in a relative increase in the importance of the dismounted soldier in future military operations. The cluster of technologies generally referred to as Virtual Environment (VE) technology has the capability to integrate the dismounted soldier into DIS. While the concept is not new (Gorman, 1990), very little progress has been made toward its implementation.

Definitions

A Virtual Environment is a simulated space with which a viewer interacts. In most current iimplementaitons, a physical simulation of a vehicle, such as an aircraft or tank, serves as the interface between the individual in the simulation and the virtual environment. In order for an individual to interact directly in a virtual environment, some or all of the following conditions must be met: free motion of the eyepoint within the space; three-dimensional, real-time interactive graphics, with stereopsis if needed; multiple senses beyond visual (e.g., audition, touch); direct manipulation of objects; and multiple interacting, mutually visible, humans.

Individual Combatant Simulation (ICS) technology is the technology necessary to represent individual soldiers in VE. Technology elements include: visual and auditory displays, sensing head and body position and orientation, speech recognition, tactile and force feedback displays, representation of whole body movement, and biomechanical articulation of dismounted soldier models (Levison and Pew, 1993). The state of the art in these areas has been reviewed by Durlach, Pew, Aviles, DiZio, & Zeltzer (1992). While the advancement of technology in these areas is outside the scope of our organizational mission and our expertise, the determination of the technological requirements to meet training objectives, the development of strategies for using VE for training, and the assessment of VE training effectiveness are all appropriate areas for behavioral science research.

OBJECTIVES AND APPROACH

Our overall VE research objective is to improve the Army's capability to provide effective, low cost training for Special Operation Forces and Dismounted Infantry though the use of VE technology and ICS. Our approach includes: a focus on the requirements for leader and individual accomplishment of unit tasks; determination of the necessary characteristics of VE technology, including fidelity requirements, that are necessary for successful training; and evaluation of the transfer of ICS training to the real world. We have established a goal of demonstrating a visual and auditory ICS interface for the dismounted soldier by October 1994.

Our approach to achieving this objective is multifaceted, and includes cooperating with the Naval Training Systems Center on a review of the state of the art in VE Technology (Durlach et al, 1992) and its applicability to ICS (Levison and Pew, 1993); and an assessment of dismounted unit ARTEP tasks in terms of their supportability in VE (Jacobs et al., 1993). However, the key element in our program is the Virtual Environment Research Laboratory.

The Virtual Environment Research Laboratory was established by a contract between ARI and the University of Central Florida Institute for Simulation and Training (IST) in July 1992. It is located at IST and uses their facilities and equipment. In carrying out research in the laboratory, ARI personnel (research psychologists) plan experiments, develop the specifications for the environments and interfaces, conduct the experiments, analyze the data, and report the results, while the IST personnel (computer scientists) configure and develop the necessary hardware and software to conduct the research.

The overall scheme for the research is shown in Figure 1, the Virtual Environment Research Pyramid. The figure shows our research plan as sequential progress up the levels of the pyramid. At the ground level are the task requirements for dismounted soldier training, as documented in Jacobs et al. The next level represents previous research in the use of VE for training. This is not a thick layer of the pyramid. When we began our research, we found only one article (Regian, Shebilske, & Monk, 1993) on the use of VE for training.

We began our experiments at the third level, which has three distinct elements. The first, psychophysical capabilities, is concerned with how well available technology enables individuals to see and hear in a VE. The second, psychomotor capabilities, is concerned with their skills in performing simple tasks. The third, comfort, convenience, and side effects, is concerned with the impressions and side effects of exposure to VE. The research reported in this paper was conducted at this level in the pyramid.

At the fourth level is research concerned with use of VE to teach spatial knowledge, particularly the configuration of and routes through large buildings. While there is some research in this area, the use of a virtual building model has never been compared with use of the actual building as a means of training, nor have different strategies for the use of VE been compared. We have two experiments planned in this area.

At the fifth level is the use of VE to represent exterior terrain, both for training land navigation tasks, and for applying land navigation skills in the conduct of mission rehearsals and combat simulations. At the sixth level is the use of VE for tasks which involve situational awareness, i.e, complex tasks performed in a changing environment, such as

searching a building for a moving object. The seventh level, team situational awareness, involves the same tasks, but performed by teams rather than as individuals, so that communication and cooperation among team members is required.

VIRTUAL PRESENCE

For our purposes, virtual presence is the experience of being in one place when you are physically in another. The strength of this experience is often referred to as the sense of "immersion" that VE provides. Presence could be a valuable concept for enhancing training if it could be shown that those factors which enhance a sense of presence also improve training effectiveness and transfer (Sheridan, 1992). Witmer and Singer (in preparation) have suggested that the extent to which an individual experiences presence is related to two types of factors: individual susceptibility, and the characteristics of the VE itself. They have developed two questionnaires on this conceptual basis: a susceptibility questionnaire to measure an individual's susceptibility to immersion, and a post-immersion questionnaire to assess the extent to which the individual was immersed during the experience. We are using both questionnaires in our research.

THE VIRTUAL ENVIRONMENT PERFORMANCE ASSESSMENT BATTERY

The Virtual Environment Performance Assessment Battery is a set of tasks and performance measures developed to assess human performance and the effects of immersion in the VE as a function of training and system characteristics. The battery serves several important functions. First, the tasks provide a means to bring research participants to a basic level of proficiency in prerequisite VE skills (e.g., locomotion, manipulating objects). Thus students can learn the techniques necessary to "walk" through building models before a specific building model is used to teach them routes through that building. Second, the tasks can be used as "behavioral benchmarks" for interface hardware and software comparisons, to determine quickly the effects of system changes (display resolution, update rate) on task performance. Third, task performance can provide statistical controls for future research. Finally, they provide a baseline for investigating side effects, such as simulator sickness.

The tasks were selected to be components of what soldiers would do in VE, and to some extent "look like" military tasks, but require no military training. This permits the use of college students as research participants. Five task categories were derived from the simple concept that soldiers move, communicate, and employ weapons: vision, locomotion, manipulation, reaction time, and tracking. Brief descriptions of all tasks are provided in Table 1.

We have conducted two experiments using some or all of these tasks. The first examined the sensitivity of task performance to different control devices and limited practice, and provided data on the occurrence of simulator sickness. The second used a subset of the tasks to examine the effects of extended practice on both performance and simulator sickness. The hardware and software used for both experiments were the same. The tasks were presented using two 486/50mhz PCs with Intel DVI display boards, a Virtual Research Corporation Flight Helmet, a Polhemus Isotrak head tracker, and either a Gravis Joystick or a Spaceball Tech Spaceball. Sense8 WorldToolKit software, with a parallel option to connect the two PC's, was used.

Experiment 1

Table 1.

Virtual Environment Performance Assessment Battery Tasks

TASK CATEGORY	TASK NAME	TASK DESCRIPTION
Vision	Acuity	A Snellen eye chart
	Color	Ishihara color plates
	Distance Estimation	Indicate when the image of a human figure, moving toward the viewer from a distance of 40 feet, is 30, 20, 10, 5, and 2.5 feet away.
	Search	From a seated position in the center of a 20 x 20 x 20 foot room, locate a moving ball initially not within the field of view.
Locomotion	Corridor (Intro)	Move down a straight corridor to a target location, turn around, and return to the starting point.
	Back-up	The same as the Corridor task, except move backwards to the starting point without turning around.
	Turns	Move through a series of corridors connected by 10 alternating left and right right-angle turns.
	Figure 8	Move through a figure-8 shaped corridor.
	Doorways	Move through a series of rooms connected by doorways, offset so that a curved course must be followed.
	Windows	Like doorways, except that some of the openings are elevated, so that vertical, as well as horizontal, movement is required.
	Elevator	Move forward through a structure while going over or under a series of vertical obstacles.
Manipulation	Slide	"Grasp" an object and move it horizontally to a target location.
	Dial	"Grasp "a dial and rotate it to a target orientation.
	Bins	"Grasp" a ball located in one of three rows of three bins each, and move it out of the original bin and into a target bin.
Reaction Time	Simple	Indicate when a "X" appears at a designated spot on the display.
	Complex	Indicate in which of four boxes an "X" has appeared.
Tracking	Head Control, Stationary Target	Using head movements, move a cursor centered in the viewing device over a stationary target.
	Head Control, Moving Target 1	Using head movements, move a cursor centered in the viewing device over a target moving in a single straight line.
	Head Control, Moving Target 2	Using head movements, move a cursor over a target moving in a path which includes a single turn.
	Device Control, Stationary Target	Using a control device, move a cursor over a stationary target.
	Device Control, Moving Target 1	Using a control device, move a cursor over a target moving in a single straight line.
	Device Control, Moving Target 2	Using a control device, move a cursor over a target moving in a path which includes a single turn.

Procedure. Twenty-four research participants each completed the first twenty tasks shown in Table 1. Participants were primarily college students who had normal or corrected-to-normal vision and were paid for their participation. One-half of the participants performed the tasks using a spaceball as the control device, and the other half used a joystick. The tasks were performed in two sessions on separate days.

At the end of each session, participants completed the Simulator Sickness Questionnaire (SSQ; Kennedy, Lane, Berbaum, and Lilienthal, 1993) and the Post-immersion Presence Questionnaire. The SSQ is a 16-item questionnaire on which participants report some symptoms on a four-point scale (None, Slight, Moderate, or Severe) and others as being present or absent. It produces a Total Severity score and three subscale scores: Oculomotor (eyestrain, difficulty focusing, blurred vision, headache); Disorientation (dizziness, vertigo); and Nausea (nausea, stomach awareness, salivation, burping).

Results. The most striking results are shown in Figure 2, which shows mean completion time per segment for each of the locomotion tasks as a function of Control Device. The difference between the two groups was significant for each task ($p<.03$). A similar pattern of completion times was found for the manipulation tasks (see Figure 3). Again all control device differences were statistically significant ($p<.02$). The pattern did not hold for tracking tasks. There were no differences between head and device tracking, or between spaceball and joystick. There were consistent significant practice effects for all manipulation tasks ($p<.05$), but for only two locomotion tasks, Intro (the first locomotion task performed) and Windows (the first locomotion task performed which required participants to "fly").

Of the 24 participants, one became too ill to complete a session. Her data are excluded from Figure 4, which shows the results of the SSQ administration for both of our experiments. Zero on any scale represents the complete absence of symptoms. The first two clusters of histograms show the results of the administrations following Session 1 and Session 2. Our participants reported more severe symptoms than Kennedy et al's (1993) aviators. Symptoms were also significantly more severe following Session 2 than Session 1 ($p<.05$ for each subscale). Nevertheless, no participant reported "severe" symptoms of any kind. Seven of 19 reported "moderate" eyestrain as the worst symptom. A majority of the participants reported slight or none to each symptom.

The presence questionnaires were found to have satisfactory internal consistency (.74 for the susceptibility scale and .74 and .87 for the first and second administrations of the post-immersion questionnaire). Susceptibility did not predict experienced presence. Experienced presence was negatively correlated with simulator sickness ($r = -.45$ for session 1 and -.46 for session 2, $p < .05$ for both): the higher the Total Severity score, the lower the amount of presence experienced.

Discussion. The objective for the first experiment was to determine if the VEPAB tasks were sensitive to differences in control devices and to practice effects. The Locomotion and Manipulation tasks showed sensitivity to control devices, but the Tracking tasks did not. We suspect that the slow system update rate for those tasks (about 300 ms.) made them so difficult that they could not be performed well with any control device. Overall, participants were able to keep the cursor on the moving target less than 9% of the time.

Whether or not the tasks are sensitive to practice effects is less clear. Only the manipulation and some locomotion tasks showed practice effects. We expect that this is because participants had little time for practice.

The SSQ data show that simulator sickness is an aspect of VE that must be taken seriously, but it is not a "show stopper." Despite the limits that we placed on exposure (breaks approximately every 20 minutes and total exposure less than one hour per day), we still found that most participants reported some symptoms. There is a lack of other data to use for comparison. We do not know, for example, what symptoms our participants would have reported prior to the start of a session, or after a similar period of time spent word processing. We do not know if there are behavioral consequences of the exposure. For example, is balance affected? We do know that our participants reported more severe symptoms than Navy aviators did following simulator use; but then, Navy aviators are very different from college students.

There are several reasons why reported symptoms might have been more severe after the second session than after the first. The amount of time spent performing tasks in VE was longer in the second session, and this may account for the difference. The SSQ may be a "reactive" measure, that is, completing it once, after the first session, may cause the participant to be more aware of their symptoms during the second session. There may be a cumulative effect of repeated exposures. Finally, task differences between the first and second sessions may have contributed. Both tasks that involved "flying," or vertical movement, were in the second session.

Experiment 1 left us with several questions. How much improvement could we expect with additional practice, and how rapidly would participants reach some sort of plateau? Would extended practice eliminate the difference between the Spaceball and Joystick groups? What is the normal level of occurrence of SSQ symptoms, without exposure to VE? Did the severity (relative to Naval aviators) of the simulator sickness symptoms indicate that there were corresponding vestibular disturbances? To answer these questions we conducted a second experiment which involved extended practice on a subset of VEBAP tasks.

EXPERIMENT 2

Procedure. Six ARI employees with normal or corrected-to-normal vision performed each of five tasks from the VEPAB in 11 experimental sessions on six different days (one session on the first day and two on each of the remaining five). The order of the tasks was counterbalanced across sessions. They were: Turns, Figure 8, Windows, Bins, and one tracking task (Device Control, Moving Target 2). The Turns, Bins, and Windows tasks were the same as those used in Experiment 1. The Figure 8 task was modified so that it ran continuously for five minutes, rather than stopping after two complete circuits. Three participants performed the tasks using the joystick, and three using the spaceball. Participants completed the SSQ prior to and after their daily practice. As an additional measure of simulator sickness, we measured postural stability before and after each participation. This was accomplished by having each participant stand on their non-preferred leg, with their arms crossed over their chest and their eyes closed while wearing the Flight Helmet. The time that they could sustain this position was measured by an observer, and head orientation was recorded approximately eight times per second by the Pohlemus Isotrak.

Results. Task performance is summarized in Figures 5, 6, and 7. Regression analysis showed a significant improvement on all tasks

with practice ($p<.05$). On the Turns and Windows tasks, most of the improvement occurred in the first few sessions, while on the others it appeared to be largely linear. Also on the Turns and Windows tasks, practice greatly reduced the differences between the Spaceball and Joystick groups while on the Bins and Figure 8 tasks they remained nearly constant. Again, the tracking task proved to be extremely difficult, no matter what control device was used, although a practice effect was evident.

The results of the SSQ administrations are shown along with those of Experiment 1 in Figure 4. (There were no meaningful differences across days, so the figure shows averaged pre-exposure scores across days, and post-exposure scores for the first and last days, which were representative.) Clearly, our participants were not free of simulator sickness symptoms prior to exposure to VE. The overall level of post-session symptoms reported was slightly lower than those reported by the participants in experiment 1 after their first session, but did not show any cumulative effect, nor on the other hand, was there a noticeable adaptation.

The results of the test of postural stability are shown in Figure 8. There appears to be some slight practice effect, but no decline in stability as a result of exposure to VE. Our analysis of head orientation during this test showed no discernable patterns.

Discussion. We have concluded that the subset of VEPAB tasks we used for this experiment are sensitive to practice effects. While the tracking task is particularly difficult, and the turns task is relatively easy, none of the tasks are so difficult or easy that performance does not improve with practice. We have also concluded that the joystick is preferable to the spaceball for use in our future experiments. While spaceball users eventually performed about as well as joystick users on some tasks, for other tasks the joystick produces superior performance after even extensive practice (55 minutes in the case of the Figure 8 task). This should not be taken to indicate that the spaceball is an inferior control device. We suspect that our participants were more familiar with the joystick than the spaceball prior to the experiment. Joysticks are a common component of video games: spaceballs are not. Also, our system updated relatively slowly (approximately 3 to 9 times per second, depending on the task). Since the spaceball provided little inherent feedback, this may have interacted with the slow update to make it particularly difficult to learn to use. The same result might not be obtained if the effects of applying force to the spaceball were immediately apparent.

With regard to simulator sickness, we believe it is valuable to assess participant symptoms prior to, as well as after, each experimental session. We did not find any evidence of an increase in simulator sickness as a result of repeated exposures, but neither did we find any evidence for adaptation. However, our sample was very small and unlikely to uncover any but the largest effects. We also did not find any changes in postural stability due to exposures. We are still seeking an explanation for the relatively high level of symptoms encountered in the second session of the first experiment.

FUTURE RESEARCH

These two experiments have produced a set of tasks which we can use for future experiments and a set of experimental procedures we can use for conducting that research. We have also collected baseline data regarding human performance and simulator sickness in virtual environments. This provides

the necessary basis for the conduct of experiments which are more directly involved with the use of VE for ICS.

As of this writing we have just completed data collection in one additional experiment, and have made preparations for a second. The experiment just completed compares three media for rehearsing routes through an office building: pictures and written directions; a virtual office building; and the actual building. Since it is an actual building, we will be able to test how well knowledge acquired in the virtual building transfers to the real world. The experiment which we are prepared to conduct will compare VEPAB task performance with three visual display alternatives (monitor, high resolution boom, and low resolution head-mounted display). Following those experiments, we will move to higher levels of the pyramid shown in Figure 1.

REFERENCES

Durlach, N.L., Pew, R.W., Aviles, W.A., DiZio, P.A., & Zeltzer, D.L. (1992). Virtual Environment Technology for Training (VETT) (BBN Report No. 7661). Cambridge, MA: Bolt, Beranek, and Newman, Inc.

Gorman, P. (1990). Supertroop via I-Port: Distributed simulation technology for combat development and training development. Alexandria, VA: Institute for Defense Analysis.

Jacobs, R.S., Crooks, W.H., Crooks, J.R., Colburn, E., Fraser, R.E., Gorman, P.F., Madden, J.L., Furness, T.A., & Tice, S.E. (1993). Behavioral Requirements for Training and Rehearsal in Virtual Environments (ARI TR). Alexandria, VA: U.S. Army Research Institute. In preparation.

Kennedy, R.S., Lane, N.E., Berbaum, K.S., and Lillenthal, M.G. (1993). A simulator sickness questionnaire (SSQ): A new method for quantifying simulator sickness. International Journal of Aviation Psychology.

Levison, W.H., & Pew, R.W. (1993). Use of Virtual Environment Technology for Individual Combat Simulation (TR 971). Alexandria, VA: U.S. Army Research Institute.

Regian, J.W., Shebilske, W.L., and Monk, J.M. (1992). A preliminary evaluation of virtual reality as an instructional medium for visual-spatial tasks. Journal of Communication. 42 (4).

Sheridan, T.B. (1992). Musings on telepresence and virtual presence. PRESENCE: Teleoperators and Virtual Environments, 1, 120-125.

Witmer, B.G., & Singer, M. J. (in preparation). Measuring Immersion in Virtual Environments (ARI Working Paper). Alexandria, VA: U.S. Army Research Institute.

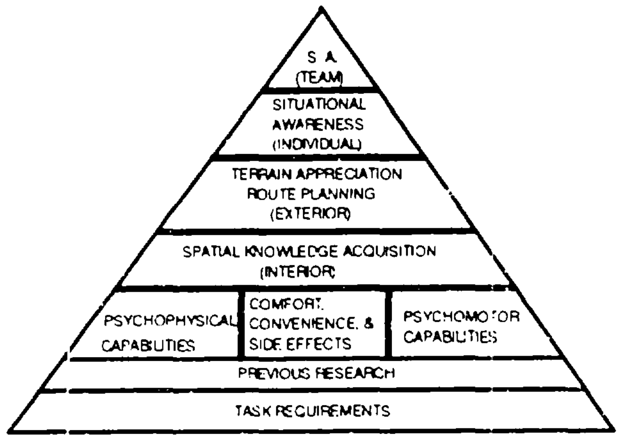


Figure 1. The Virtual Environment Research Pyramid

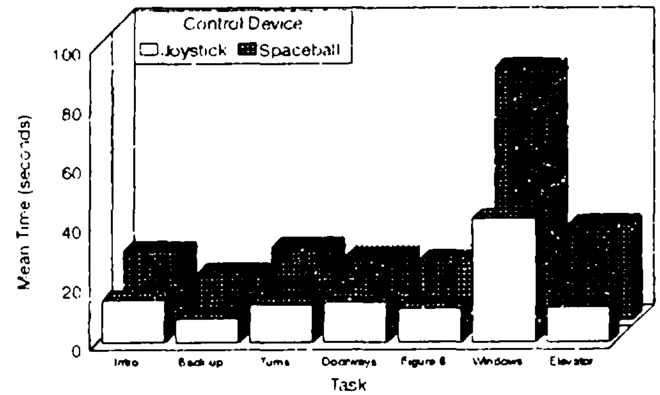


Figure 2 Experiment 1 Locomotion Task Time as a Function of Control Device

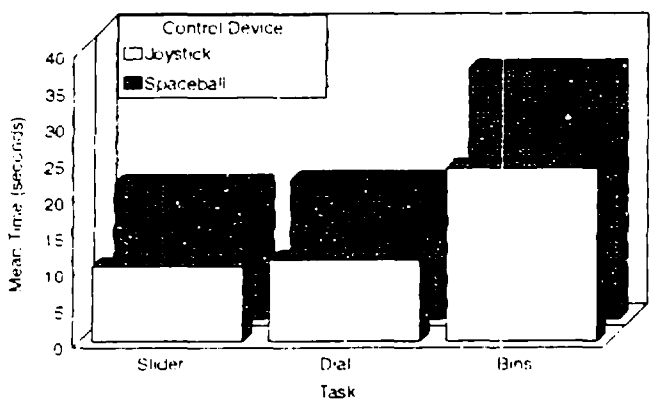


Figure 3 Experiment 1 Manipulation Task Time as a Function of Control Device

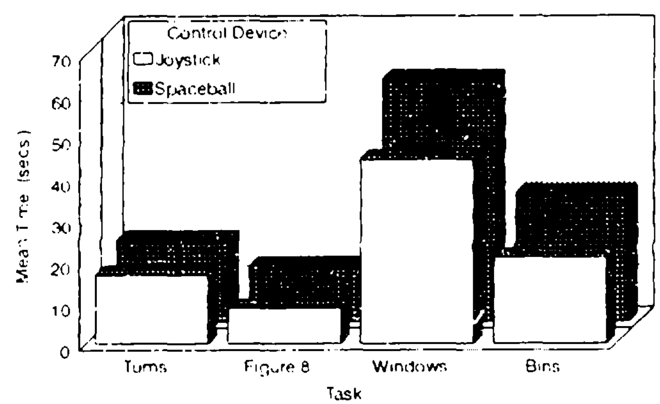


Figure 4. Experiment 2 Task Time as a Function of Control Device

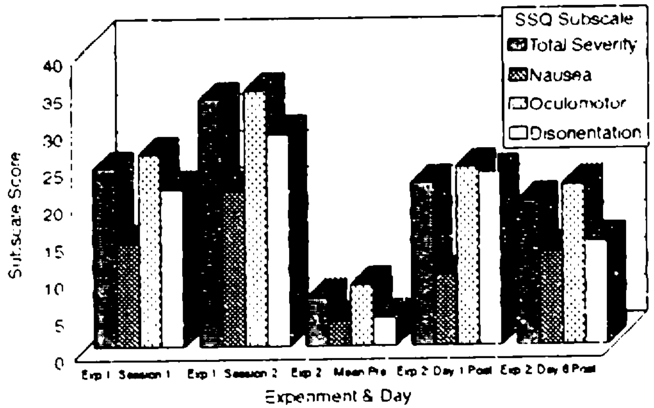


Figure 5 SSQ Subscale Scores for Experiment 1 (Sessions 1 and 2) and Experiment 2 (Pre-VE averaged over all days, and Post-VE for Days 1 and 6).

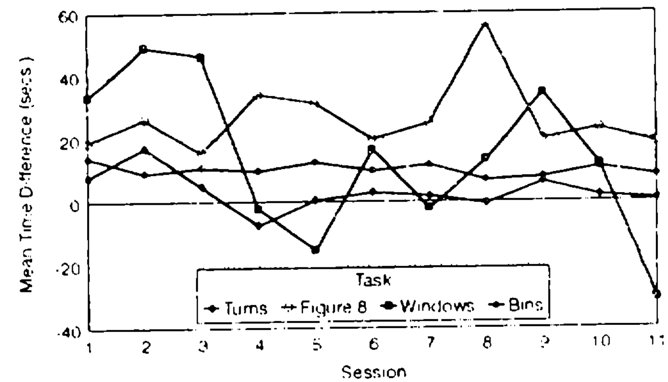


Figure 6 Experiment 2 Task Time Differences (Spaceball Joystick) as a Function of Experimental Session

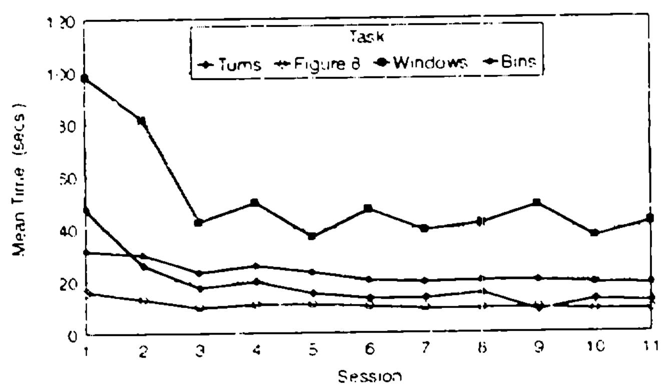


Figure 7 Experiment 2 Task Times as a Function of Experimental Session

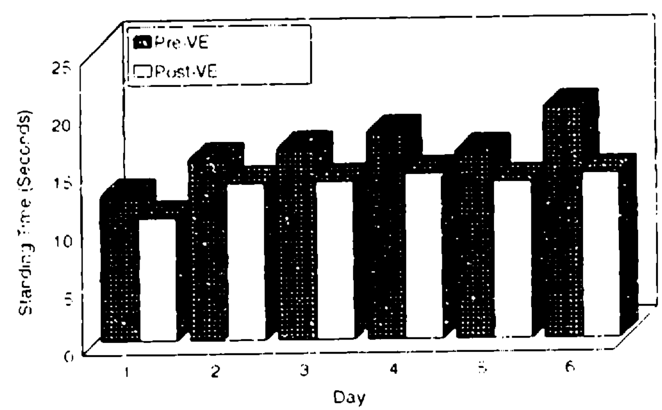


Figure 8 Experiment 2 Postural Stability as a Function of Day

APPLICATION OF A THREE-DIMENSIONAL TARGET DISPLAY FOR WEAPONS TRAINING

Albert Marshall
Edward Purvis
Ronald Wolff
Robert McCormack

Naval Training Systems Center

ABSTRACT

A 3-D large screen display for small arms and minor caliber weapons has been designed and the prototype will be displayed at the I/ITSEC 93 conference. The system provides interactive stereoscopic images of the environment and targets that virtually leap from a 100 inch diagonal video projection screen. The system uses switched LCD glasses, worn by the trainee, to convert video recorded from two separated video cameras and stored on video disk to 3-D like images. Three -Dimensional computer graphics objects are added to portray tracers and objects flying at the trainee. The prototype has been tested and the efficacy of the prototype is discussed. The use of a small motion platform with the system is also discussed.

ABOUT THE AUTHORS

Mr. Albert H. Marshall is a Team Leader / Electronics engineer in the Systems Integration Branch at the Naval Training System Center. He has specialized in developing weapon fire simulators using lasers, electro-optics, microcontrollers and video disk technology for twenty six years. He holds thirty U.S. patents. Mr. Marshall has a Masters Degree in both Physics and Electronics Engineering from Brown University and the University of Central Florida.

Mr. Edward J. Purvis is an Electronics Engineer in the Systems Integration Branch at the Naval Training Systems Center. He has specialized in developing weapon fire simulators using electro-optics, microprocessors, video processors, and graphics for nine years. He has a Masters Degree in Digital Electronics from the University of Central Florida.

Mr. Ronald S. Wolff is an Electronics Engineer in the Systems Integration Branch at the Naval Training Systems Center. He has worked extensively in Weapons Simulation Technology for the last seven years. Mr. Wolff has worked in system design and integration, analog design, electro-optics, microcontrollers, and sensor interfacing. Mr. Wolff has a Masters Degree in Electronics Engineering from the University of Central Florida.

Mr. Robert McCormack is an Electronics Engineer at the Naval Training Systems Center. He has specialized in developing weapon fire simulators. He has worked on computer program development and sensor interface to computers. Mr. McCormack graduated in 1985 from the University of Central Florida, where he is currently pursuing a Masters Degree in Computer Engineering.

INTRODUCTION

A successful training device creates a human trainer interface in which the device creates a sensory-immersing environment that interactively responds to and is controlled by the actions of the user. The creation of such an environment requires the immersion of your senses in a computer 3-D world to create the experience of actually "being there". What we hope to achieve is to create an environment real enough for you to suspend your disbelief during a period of time and make you feel you are actually facing a real world enemy. To create the desired sensory environment a large screen 3-D visual display system, interactive targets and computer controlled 3-D sound has been incorporated. The systems key attributes include:

(1) The environment is displayed in 3-D.

(2) A method to interactively remove aggressor targets which are hit as a training scenario progresses.

(3) A method which allows aggressor targets to engage and disable a trainee who does not take appropriate cover.

(4) A 3-D sound system.

Many simulator-based team trainers currently use technology which restricts realism in tactical training situations. It is anticipated that adding the 3-D will help to make the screen seem to disappear and make the trainee feel he is in the same environment as his enemy.

The prototype system developed at the Naval Training Systems Center will allow a trainee to practice and rehearse close combat training exercises such as SWAT operations with an unsurpassed level of realism and feedback in a 3-D environment. Typical events might include hostage rescue, security operations, shoot-no-shoot, ambush training situations and routine law enforcement operations.

Safety is also a concern during live fire training exercises. Since the trainer uses no live ammunition the dangers of an inadvertent weapon discharge or lead poisoning are eliminated entirely.

Much of the trainee performance data and feedback provided by the trainer is not available using live fire training. The prototype provides advantages over live actor force-on-force training in a number of critical areas. Reliability of scenario presentation is inherent in the simulator system, where as live actor based training introduces variability in the form of inconsistencies and other human errors. The prototype also provides extensive measurement of trainee behavior and achievement which can be used for feed-back and leads to objective fulfillment. Through controlled presentation of intelligent scenarios, the trainer provides more reliable decision making tactical situations than any current alternative.

"Build it and see what happens " is the maxim inventors have lived by for millennia. This new addition may possibly allow for effective and realistic training for military operations previously unobtainable through simulation. Will this technology allow us to better portray the real world and allow us to see and feel things not possible in the past ? We will discuss the prototype and results of the testing during the paper presentation.

DESCRIPTION OF THE SYSTEM

The system uses switched liquid crystal (LCD) glasses, worn by the trainees, to convert video recorded from two cameras and projected with different perspective views to form 3-D like images. See Figure 1. Each camera views a different perspective because the video cameras used to record a scenario are offset horizontally from each other like the human eyes. The average separation distance of a persons eye is 2.5 inches. The video scene taken with the left camera perspective is viewed by the left eye and the scene taken with the right camera perspective is viewed by the right eye. This video recording method presents the views a person sees with two separated eyes in the real world. Two different perspective viewpoints are synthesized by the brain to produce a stereoscopic 3-D like effect similar to the realism that the trainees experience in the real world.

FIGURE 1. LCD GLASSES AND SIMULATED WEAPON

The human brain is the last link in this optical system because it converts the spatially separated video data into a stereoscopic 3-D experience for the trainee. The addition of depth cues add to the realism of the projected environment.

The hardware useo for stereo video recording is shown in Figure 2. A view/record controller accepts the inputs from two genlocked cameras and converts them into a single signal. The signals from the two video cameras are stored in memory and operated on topologically to produce a side by side field format.

Each picture has 240 video lines per field in a four - fold interlace pattern. This pattern has the following sequence of fields: left (odd), right (odd), left (even), right (even), etc. The recorded signals are also indexed to allow the images to be later played back steroscopically. This signal is recorded by a single standaro S-video recorder and later edited and recorded to a video disk. Scenario data is recorded in a stereo format by using two standard video cameras. The two cameras are mounted on a common base and aligned to be parallel and the optical axes are approximately 2.5 inches apart. Video camera lenses, focal lengths f-stops and color balances must be identical. The output of the dual s-video cameras is 60 fields per second 15.75 KHz signals. The view record controller takes the standard s video from the left and right cameras and produces a side by side format for recording on a standard S video recorder.

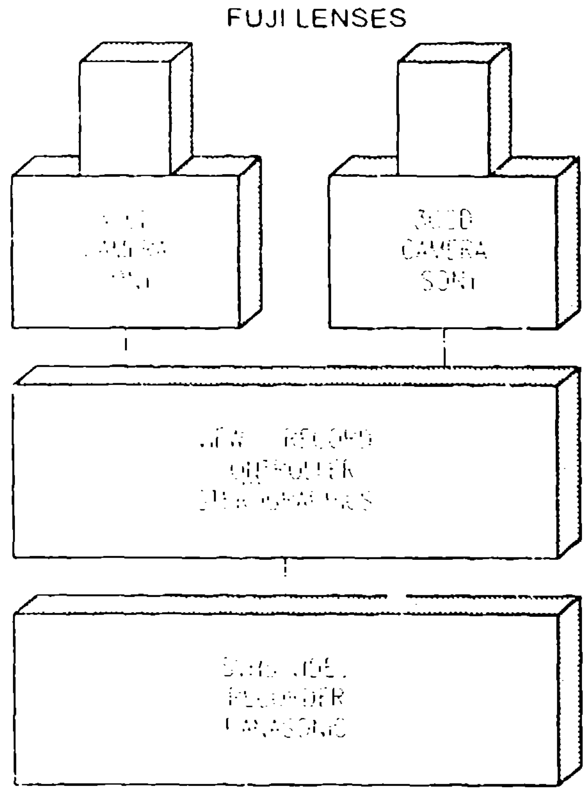


FIGURE 2. STEREO VIDEO RECORDER

The playback controller processes the stereoplexed signal to read out the sidefield lines in the odd and even sequence previously described. The video information is alternately projected at a 120 fields per second on the video screen. The Stereo Video playback block diagram is shown in Figure 3. If the trainee looks at the screen without shuttered glasses he sees what appears to be a double image. Using shuttered lenses the trainees left eye sees only the left image and the right eye the right image. Each successive field alternates from the left eye to the right eye. One eye is alternately shuttered while the other eye views the screen. The electro-optic shutters use switched liquid crystal lens materials that alternately render one lens clear and the other lens opaque.

Color, brightness and geometry of the two projected video perspectives must be identical or "eyestrain" results.

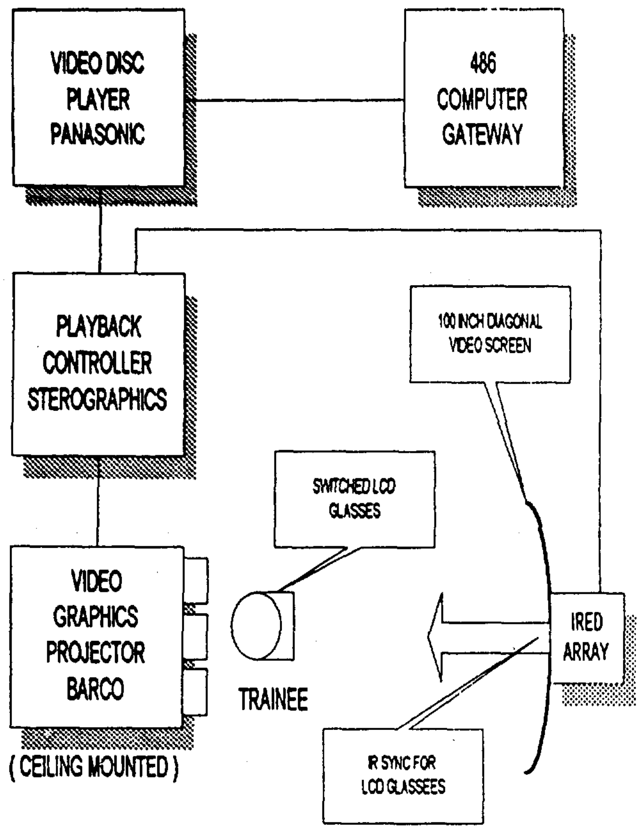


FIGURE 3. STEREO VIDEO PLAYBACK

The 120 Hz field sequential video rate is twice the customary 60 Hz field video frame rate. If alternate stereo information was recorded at alternate 60 Hz frame rate objectionable flicker would occur. However, operation at a 120 Hz frame rate requires that the glowing phosphors used in the TV projection tubes no longer emit light prior to the next field being projected. Special low persistence phosphors are used in the TV projector selected. Incomplete isolation of the right and left eye information will cause cross-talk between the shuttered eyes. Cross-talk appears to the trainee as ghosting.

Three-Dimensional computer graphics objects are also added to portray tracers in space and exploding objects flying toward the trainee. The scenario data is stored on video disk. The video projection screen displays both recorded video targets and graphics overlays using a video projector and video disk player under computer control.

Moving through the environment for a single trainee is simulated using a tread mill located in front of the projection screen. The switched LCD glasses, video recording and playback controllers are made by SteroGraphics. See Reference 1.

Each trainee has a weapon that is equipped with a collimated source of infrared (IR) energy, an infrared emitting diode (IRED). The collimated infrared source is aligned with the trainee's weapon and places an eye-safe infrared spot on the video projection screen corresponding to the location the trainee is pointing his weapon. Figure 4 shows the infrared spot tracker imaging diagram.

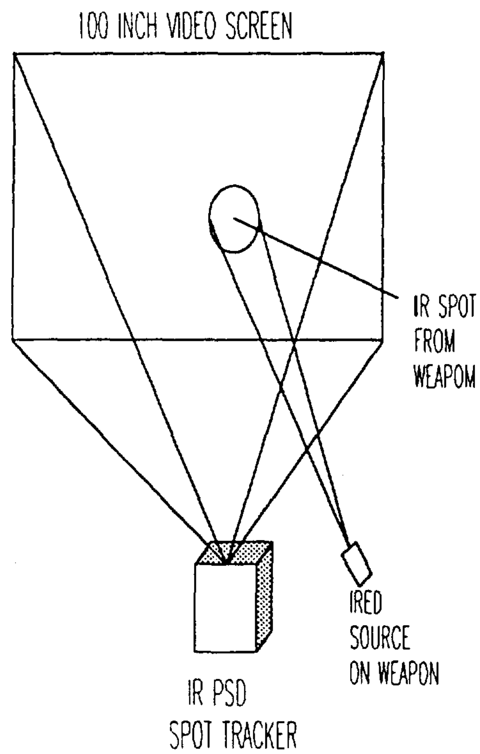


FIGURE 4. INFRARED SPOT TRACKER IMAGING DIAGRAM

The infrared sources are sequentially modulated in a time-multiplexed mode by the system computer to both identify the active weapon and to improve signal detection. A high-speed, low cost, infrared spot tracker determines the continuous X and Y position coordinates of each weapon. The optical system for the infrared spot tracker (IST) views the entire video projection screen. The infrared spot imaged onto the projection screen surface is optically transferred or reimaged to a corresponding location on a Position Sensing Detector (PSD). The system computer determines the position coordinates of the infrared spot on the PSD and consequently the

video projection screen as well. The high-speed PSD-based infrared spot tracker, generates the continuous position coordinate data of each weapon in less than 3 milliseconds; in contrast, a typical CCD-based tracker would require over 16 milliseconds. See References 2 and 3.

Once the system computer knows the position coordinates of a weapon, it can compare that data to the stored coordinates of active targets on the projection screen at the time of trigger pull. If the IST position data matches the coordinates of a target on the projection screen, a kill, wound, or miss is recorded for that weapon. The ability to have targets disappear or branch after they have been hit is very important in a trainer. Trainees are encouraged to take sensible cover as they would in the real world while engaging targets displayed on the video projection screen.

Each trainee wears a modified Multiple Integrated Laser Engagement System (MILES) type torso harness containing infrared detectors and an audio alarm device to indicate if he has been killed or wounded by an on-screen aggressor. The on-screen aggressor shoot-back is simulated by using an array of infrared emitting diodes (IREDs) located adjacent to the video projection screen. Each IRED is pointed to a particular sector within the training exercise area so that all exposed areas are exposed to shoot-back by the on-screen aggressors. The individual IREDs are turned on and off by the system computer corresponding to where the on-screen aggressor is pointing his weapon when he fires at a trainee. If a trainee does not take cover while in the field-of-fire of the on-screen aggressors he will be illuminated with infrared energy.

The infrared detectors positioned on the MILES type torso vest will detect the incident IR energy and activate an alarm to indicate that the trainee has been shot by the on-screen aggressor. Once a trainee has been hit he is considered dead and his weapon is disabled. When the trainee has been killed he will hear the alarm and his weapon is automatically disabled. After a training session is over, the video scenario is played back in slow motion. The system computer shows the continuous pointing location of each weapon by graphically displaying color coded icons representing the continuous tracker position data stored by the system computer during the actual training session. Hit, wound and miss shot locations are indicated by changing the color of the icons. The instructor can see how each trainee is handling the weapon by observing the icons during play-back.

A digital sound system is used to simulate the actual acoustical training environment of each scenario. A digital sampler digitizes, stores and plays back the background sounds as well as the synchronized gun shot sounds corresponding to the trainees and the on-screen aggressors. The sampler is under the control of a Musical Instrument Digital Interface (MIDI) port interfaced to the 486 computer for the proper timing and synchronization.

The system contains a 486 computer. The computer controls the communications to the trainee and the infrared spot tracker located in front of the video projection screen. The computer also is used to control the video projector and video/graphics adapter.

A Truevision VISTA graphics board allows a programmer to manipulate the projected video display. A frame-grabbed still video image can be displayed as background. Stored video from the video disc can be displayed in several windows which can be opened or closed as the targets are hit. These windows can be opened in either a frame-grabbed or graphics background. Graphics can be displayed overlaid on live video or on a frame-grabbed or graphics background. Sections of the image, either frame-grabbed or graphics, can be moved or copied anywhere in the image; or to off-screen memory buffer for later use. Images can be saved to disk and retrieved later. A Panasonic optical disc recorder/player, is used for storage of the scenarios.

The Barcodata projector is modified with a short persistence green phosphor tube to accommodate the 3-D 120 HZ video.

Special software developed at NTSC uses the tracker data to determine miss, wound, or kill. The software, is used to mark the positions of the various potential targets in a scenario. This is necessary in order to allow the computer running the scenario to know where the friends and foes are located. This is very important for scoring hits and misses. The software allows you to step through the video scenario frame-by-frame and fit irregular polygons around the hit areas of the targets. Each hit area can

represent either a wound or a kill. This information, along with the target window coordinates is stored to a file to be read back during operation of the trainer. The target window is the area on the screen where the target will appear.

The sound system provides sounds of the various weapons being fired by both the trainees and their on-screen adversaries. Background sounds are generated to increase realism during a training scenario. The heart of the sound system is a digital sampler playback module. A sampler digitizes, stores and plays back sound effects under the control of a MIDI (Musical Instrument Digital Interface) port.

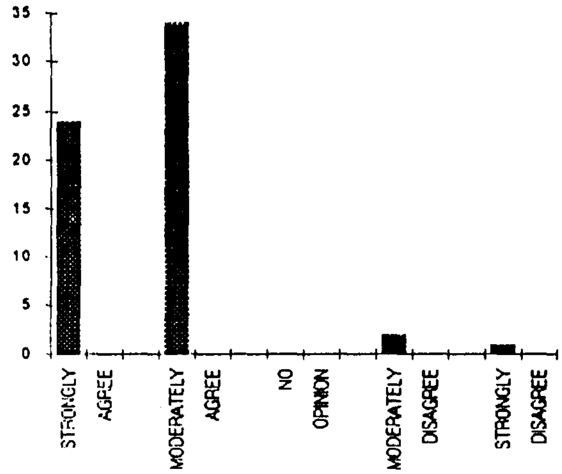
MOTION SIMULATION

Motion is currently simulated using a modified treadmill. By using the tread mill the trainee can slowly walk through the environment. A motion platform to simulate firing from a vehicle or small boat is being designed.

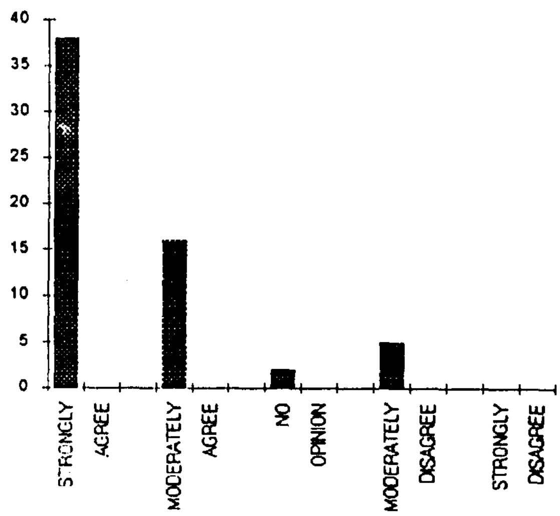
TESTING OF THE SYSTEM

The system is currently under evaluation to determine the efficacy of the 3-D prototype. The participants were asked to respond to the following statements. Initial testing evaluated 61 subjects reaction to the 3-D system. After shooting six scenarios the test subjects were asked to circle a response as strongly agree, moderately agree, no opinion, moderately disagree or strongly disagree. The results of tests are shown graphically below.

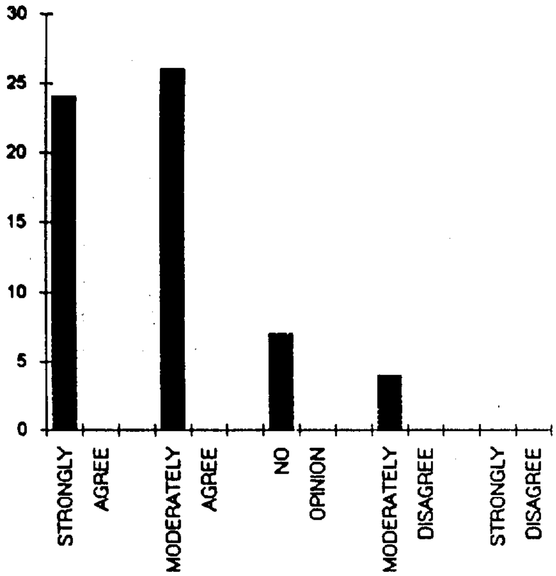
1. THE SCENARIOS APPEARED TO BE IN 3-D.



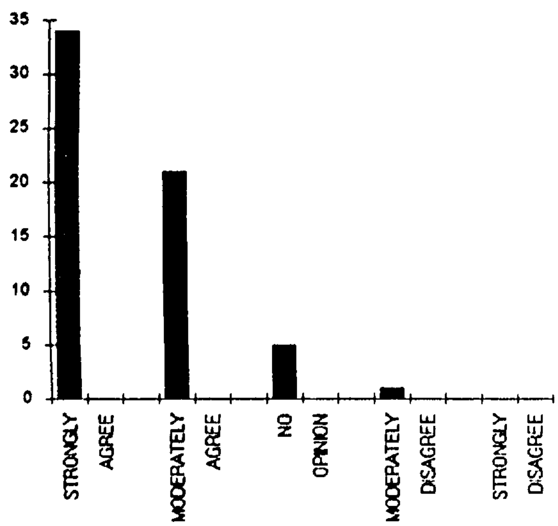
2. THE USE OF THE 3-D EFFECT MADE THE GRAPHICAL REPRESENTATIONS OF OPPONENTS MORE BELIEVABLE.



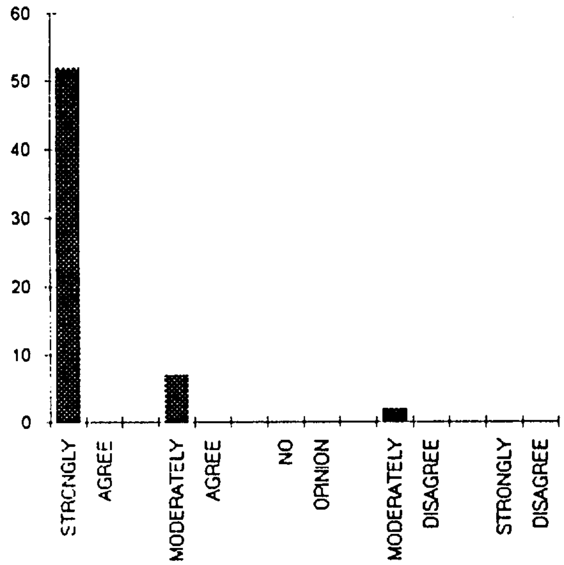
3. THE 3D GRAPHICS MADE IT EASIER FOR ME TO IGNORE MY ACTUAL SURROUNDINGS, AND THEREFORE GET MORE INVOLVED IN THE SCENARIOS.



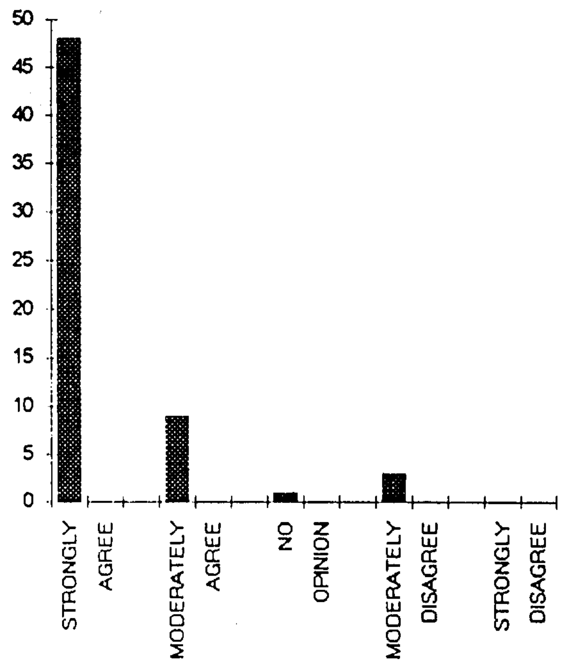
4. THE ABILITY TO GET MORE INVOLVED IN THE SCENARIOS WITHOUT DISTRACTION CONTRIBUTES TO THE TRAINING EFFECTIVENESS OF THIS TRAINER.



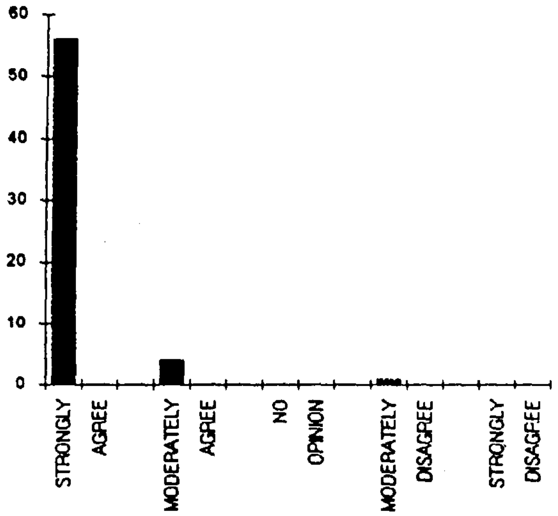
5. I DID NOT EXPERIENCE DIZZINESS DURING OR AFTER USING THIS TRAINER.



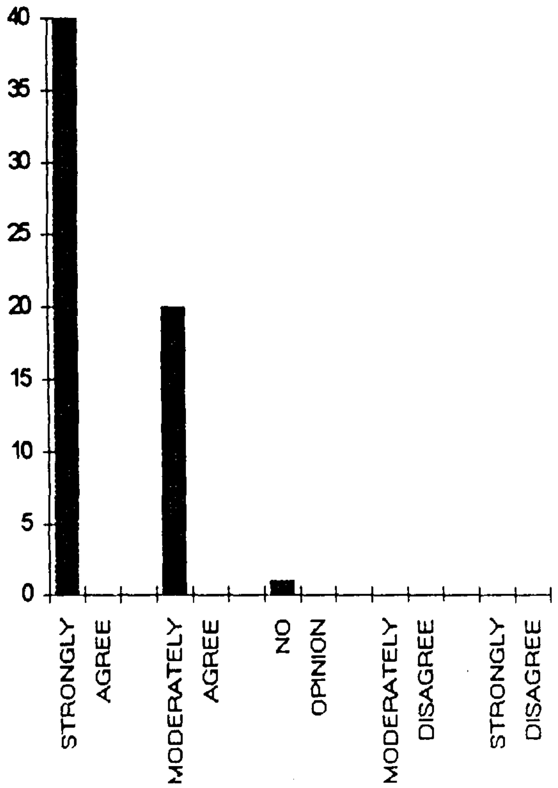
6. I DID NOT EXPERIENCE EYE STRAIN OR HEADACHE DURING OR AFTER USING THIS TRAINER.



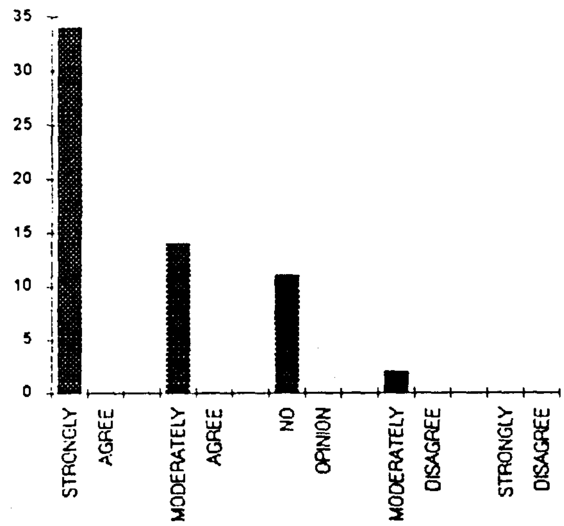
7. I DID NOT EXPERIENCE STOMACH UPSET DURING OR AFTER USING THIS TRAINER.



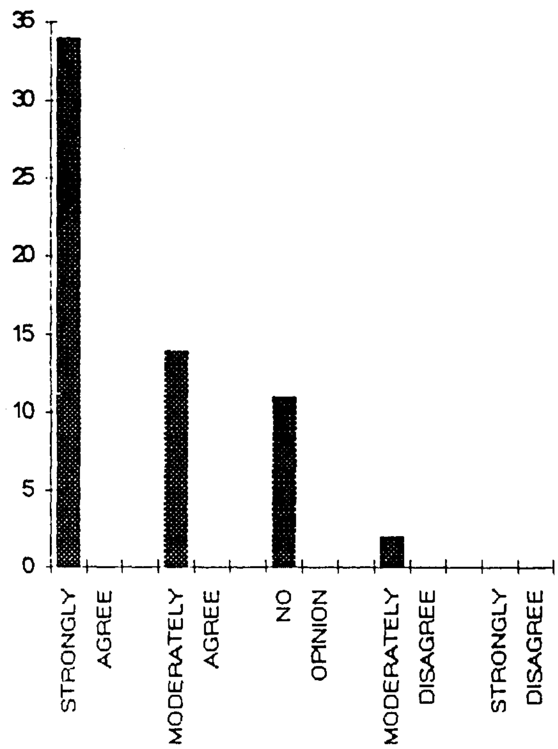
8. USING THIS TRAINER WAS AN EXCITING EXPERIENCE



9. ADDING THE 3-D CAPABILITY TO THIS TRAINER INCREASES THE COST OF IT BY APPROXIMATELY $ 10,000, I THINK THAT THE TRAINING BENEFITS MERIT THIS EXPENSE.



10. WEARING THE 3-D GLASSES DID NOT BOTHER ME OR DECREASE MY SHOOTING PERFORMANCE



RESULTS

The majority of the 61 subjects thought the addition of 3D was an exciting improvement and made it easier to ignore the actual surroundings and get more involved in the scenarios. The majority of the subjects also thought the additional cost difference over the conventional shoot-no-shoot trainer was worth it. Wearing the 3-D glasses was not deemed to bother or decrease the subjects shooting performance.

The majority of the persons that used the prototype experienced no symptoms of simulator sickness. Three of the 61 subjects had problems seeing 3-D.

CONCLUSIONS

Based upon the results of this initial testing the use of 3-D scenario presentation techniques appears to be warranted. Additional research on optimal training methods and improved 3-D display technology is continuing at NTSC. A small motion platform is currently being developed to simulate both walking and firing weapons from a moving platform..

REFERENCES

1. "The Crystal Eyes Handbook" Lenny Lipton, StereoGraphics Corp. 1991.

2. Marshall, Wolff, McCormack and Purvis. " Weapons Team Engagement Simulator", Proceedings. I/ITSC, Nov. 1990.

3. NTSC, " Weapons Team Engagement Trainer", Government Furnished Information, Dec. 1992.

ADVANCED WEAPONS TEAM TRAINING TECHNOLOGY

Robert McCormack
Albert Marshall
Ronald Wolff
Jeffrey Horey
Edward Purvis

Naval Training Systems Center
Orlando, Florida

ABSTRACT

Many simulator-based weapon team trainers currently use technology which restricts both realism and the ability for thorough team performance measurements in tactical training situations. This paper describes a training system prototype which uses new technology to improve simulation training for weapon fire teams. These new developments include intelligent video branching, location detection of trainees, interaction between trainees and their on-screen aggressors, computer networking of multiple video projection screens within multiple rooms, a wireless data communication system allowing full unrestricted mobility, a high speed weapon tracking system, and a digital MIDI controlled sound system.

The simulator developed at the Naval Training System Center will allow up to nine trainees to practice and rehearse close combat training exercises such as low intensity conflict, light infantry, SWAT, and security operations with a high level of realism and feedback. Typical events might include security operations, hostage rescue, shoot-no-shoot, outdoor squad engagements, and routine law enforcement operations in a common threat team training environment.

ABOUT THE AUTHORS

Mr. Robert McCormack is an Electronics Engineer in the Naval Training Systems Center. He has specialized in developing weapon fire simulators. Mr. McCormack has a Bachelors Degree in Electronic Engineering from the University of Central Florida where he is pursuing a Masters Degree.

Mr. Albert H. Marshall is a Team Leader / Electronics engineer at the Naval Training System Center. He has specialized in developing weapon fire simulators. He holds thirty U.S. patents. He has a Masters Degree in both physics and electronics engineering from Brown University and the University of Central Florida.

Mr. Ronald S. Wolff is an Electronics Engineer at the Naval Training Systems Center. Mr. Wolff has worked extensively in Weapons Simulation Technology including system design, analog design, and electro-optics. Mr. Wolff has a Masters Degree in Electronics Engineering from the University of Central Florida.

Mr. Jeffrey D. Horey is a psychologist at the Naval Training Systems Center. He specializes in system design, human factors, and training effectiveness evaluation. Mr. Horey holds a Master of Philosophy from George Washington University and a BS from Stetson University.

Mr. Edward J. Purvis is an Electronics Engineer at the Naval Training Systems Center. He has specialized in developing weapon fire simulators. He has a Masters Degree in Digital Electronics from the University of Central Florida.

INTRODUCTION

The need for training military and law enforcement teams in close combat tactics has been demonstrated dramatically in recent months in foreign and domestic operations. Situations requiring the extraction of personnel, expulsion of terrorists, or recovery of property challenge individual and team skills in decision-making, marksmanship, and engagement tactics. While several marksmanship training systems currently exist for training close combat skills, there are serious deficiencies in these systems which detract from their effectiveness. This paper describes an advanced training system prototype, the Weapons Team Engagement Trainer (WTET). The WTET was developed specifically to address the need for improved fidelity in weapon team trainers.

Typical small arms training systems use technology which restricts both the fidelity of the tactical training situations and the ability to thoroughly measure both individual and team performance. Current training systems suffer reduced fidelity in the following ways:

- Trainees are tethered to the training apparatus, often by bulky cables, reducing the amount of available tactical movement within the training environment.
- Trainees are not engaged or "shot at" by the on-screen aggressors in a manner which instills a sense of urgency or realism.
- Aggressor targets do not realistically react to the actions of the training team.
- Trainees do not receive complete feedback on their behavior in terms of tactical movement, weapon handling, and overall mission success.

The WTET provides a multiple room training environment in which a team of up to nine trainees engage aggressor targets displayed on video projection screens. A training team can freely move within the training environment while participating in room clearing, hostage rescue, or outdoor squad engagement scenarios. The WTET allows real-time interaction between the aggressor targets and training team. During each scenario the training team must use proper tactics and take appropriate cover to successfully complete the mission. Also, an extensive after action review of individual and team performance measures allows the instructor to identify remedial training needs.

SYSTEM DESCRIPTION

The WTET accommodates training for nine military or law enforcement trainees in a common threat scenario. The trainees interact with multiple video projection screens setup in different training rooms. Video disc players display scenario scenes and target images for training team interaction. A network of multiple computer systems control the scenario's progression based upon the tactical doctrine of the aggressor force and the real-time behavior of the training team. An extensive after action review incorporates a variable speed replay using graphical icons and messages. A separate icon is displayed for each trainee providing information on shots fired, weapon status and shot location. In addition, a video recording of the training team's movements is displayed in synchronization with the after action review.

A modular system design was used to allow system flexibility. Each video screen's target presentation is controlled by a subsystem (Video Station). An Instructor Operator Station performs real-time data collection, network control and monitoring of each Video Station. This approach allows the training environment to be reconfigurable by varying the number of training rooms and video stations within those rooms. The current prototype of the WTET uses a configuration of three video stations within two rooms. Figure 1 shows an example configuration of the training environment. An additional adjacent wall video station was added to illustrate expandability of the training environment.

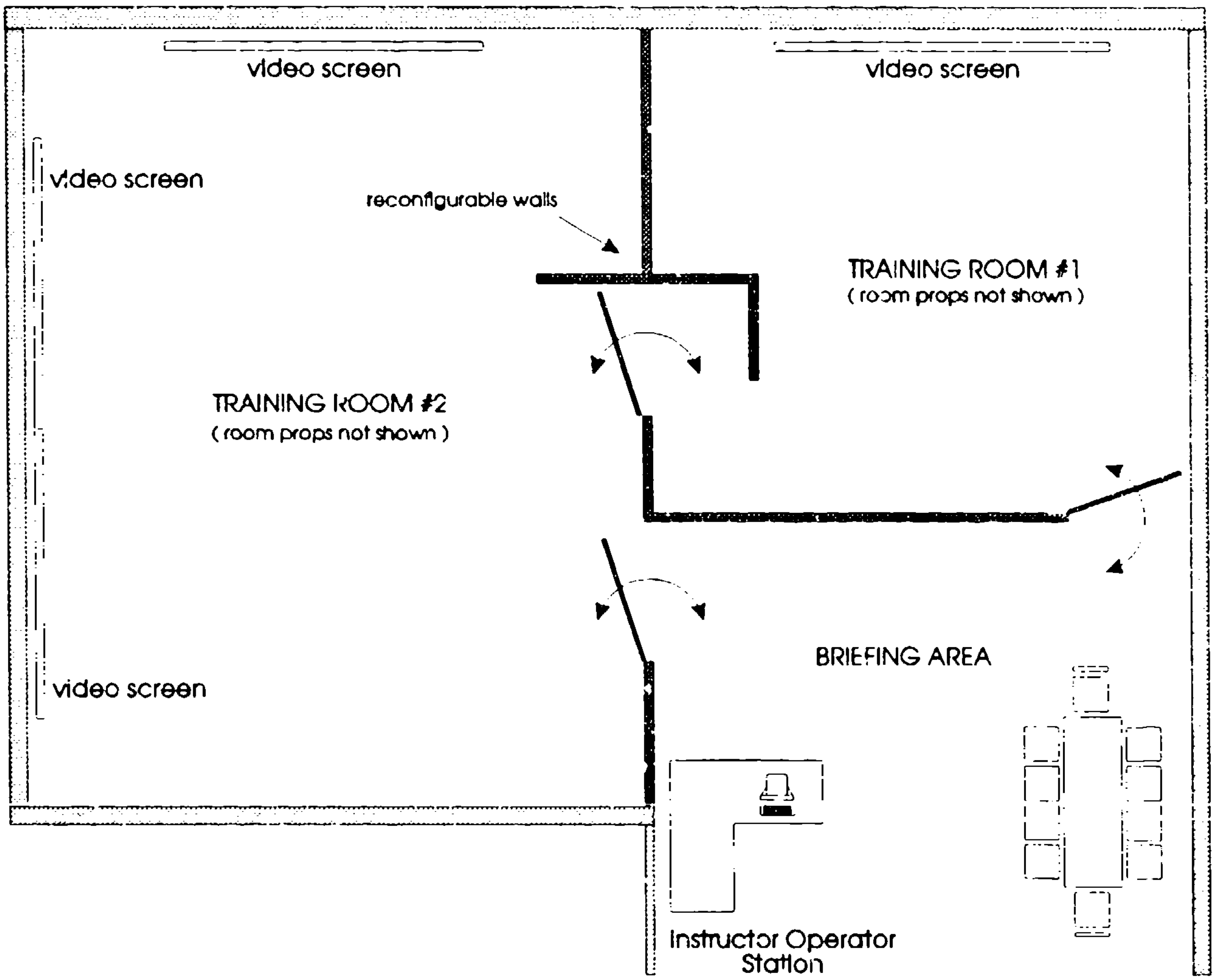


Figure 1. Layout of WTET Training Environment.

The WTET training environment is designed to allow training teams to prepare for missions, execute missions, and receive feedback on critical dimensions of performance. A briefing area allows teams to coordinate mission prebrief information. Scenario diversity is increased by incorporating reconfigurable walls and movable props in training rooms.

Each trainee is equipped with a miniature wireless communication system using RF spread spectrum technology. This RF communication system is located on a MILES type detector harness. The MILES type harness is used to determine each trainee's location and visibility. Each trainee's weapon has a barrel-mounted collimated infrared (IR) source, aligned to the weapon's sight line. The IR source places an eye-safe IR spot on the projection screen's surface. A high speed IR spot tracking system continuously determines each trainee's on-screen weapon aim point. Figure 2 shows a trainee equipped for the WTET. Trainees are able to freely move throughout the training environment while their weapon position, weapon status, and physical location are continuously monitored by the training system.

Figure 2. Trainee Wearing WTET Equipment.

SYSTEM DESIGN

The WTET system development addressed the shortcomings inherent in the technology currently used in weapon team training systems. For this reason, the WTET system design incorporates both off the shelf components and custom designed electronic subsystems. An RF communication system allows free trainee mobility. A high speed tracking system produces accurate and continuously measured weapon aim point position for multiple trainees. A trainee location system determines each trainee's location and visibility. Processing is distributed between the Instructor Operator Station (IOS) and the Video Stations. The components of the Instructor Operator Station and Video Station are shown in Figures 3 and 4.

The function of the IOS is to perform real-time data acquisition and intelligent scenario control. The IOS provides a user friendly menu system which allows the instructor to control scenario parameters. A digital parallel I/O adapter controls the RF communication system and the trainee location system. An analog input adaptor interfaces the high speed tracking system. A MIDI adapter sends sound effect messages to the digital sampler module within the sound system. An Ethernet adapter transfers data packets to the Video Stations and monitors the status of each station in real-time.

Each Video Station responds to network data and command packets. A video disc player generates scenario scene and target images. Transitions between video segments (branching) is smoothed by using the frame grabbing capability of the video graphics adapter. Instantaneous branching has been demonstrated using the combination of two video disc players and a video switcher. Also, recent developments in video disc technology allow rapid branching capability without loss of video sync signal (built in frame grab capability). A computer controlled S-VHS deck allows synchronized video recording of the trainee movements within the training environment.

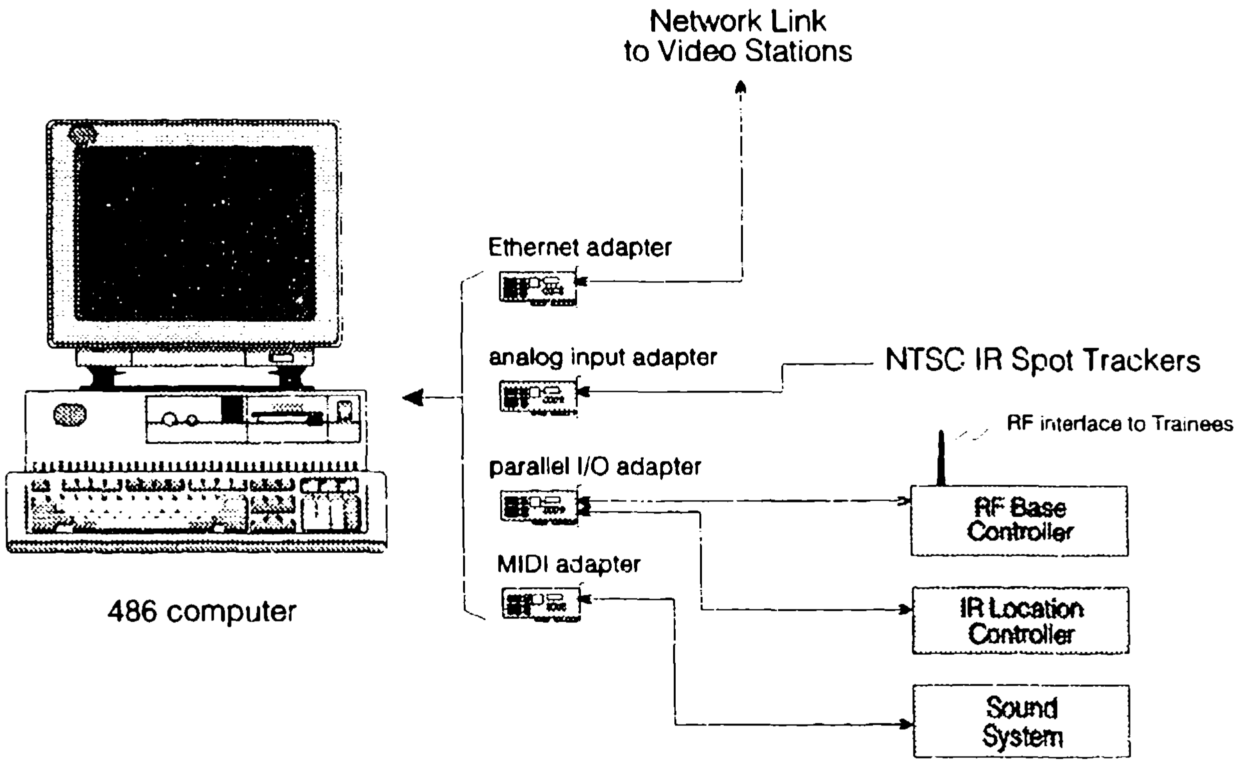


Figure 3. Components of Instructor Operator Station.

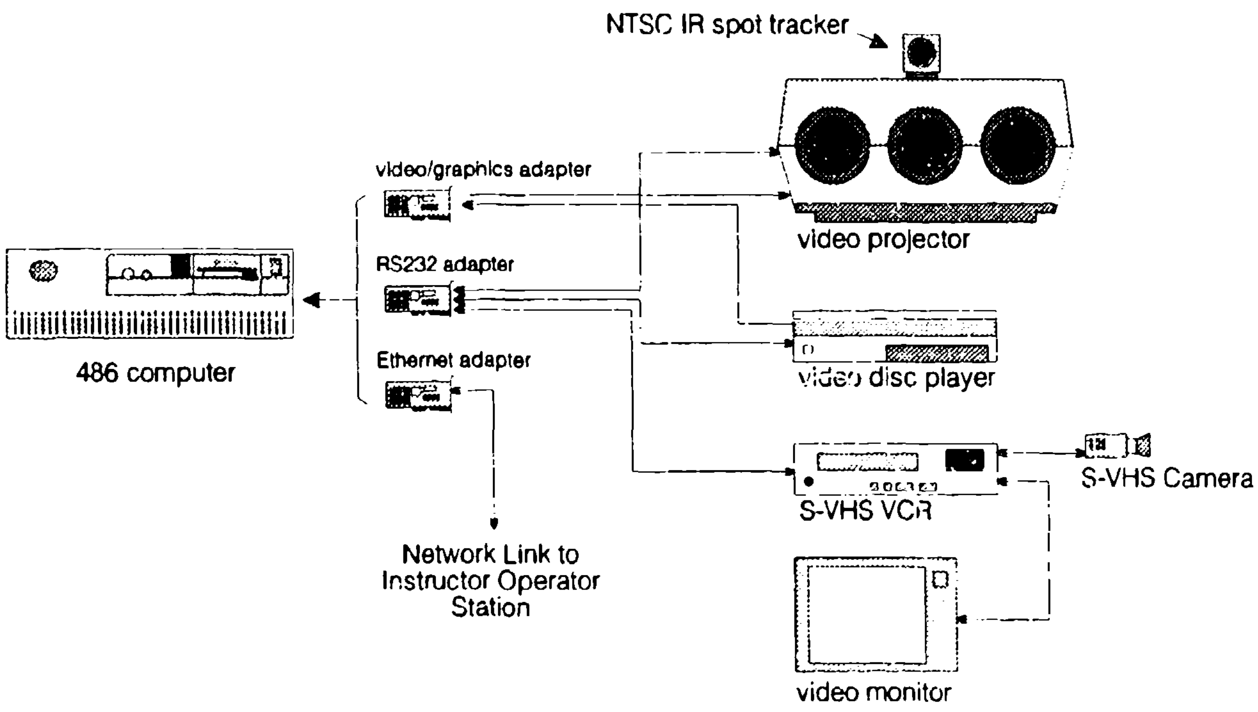


Figure 4. Components of Video Station.

Hardware Developments

The hardware developments consisted primarily of the following systems:

- High-Speed Infrared Spot Tracking System
- Trainee Location and Aggressor Shoot back System
- Wireless Data Communication System
- Digital Sound System

High-Speed Infrared Spot Tracker - To overcome the disadvantages of typical CCD-based tracking systems, NTSC has developed a low-cost, high-speed, Infrared Spot Tracker (IST) utilizing a two dimensional lateral-effect photo diode, the Position Sensing Detector (PSD) [1, 2]. The PSD is not a discrete charge transfer device, but rather a continuous analog output device. In contrast to other types of position sensing devices such as CCD detectors, the PSD offers higher resolution, faster speed, larger dynamic range, and simpler signal processing.

When a light spot falls on a PSD, an electric charge proportional to the light energy is generated at the incident position. The electric charge travels through the resistive material of the PSD and is collected by four outer electrodes. Because the detector resistivity is uniform, the photo current is inversely proportional to the distance between the incident light position and the electrodes. An A/D board in the computer is used to read the four analog tracker output voltages V_{x1}, V_{x2}, V_{y1}, and V_{y2} and calculate the position coordinates based on two simple equations.

$$X_{loc} = (V_{x2} - V_{x1}) / (V_{x2} + V_{x1})$$

$$Y_{loc} = (V_{y2} - V_{y1}) / (V_{y2} + V_{y1})$$

In operation, the IST views the entire active video projection screen area with a custom designed low F/number wide angle lens. The collimated infrared light spot from the weapon is imaged onto the screen and then re-imaged onto the PSD by the wide angle lens. The PSD and associated electronics converts the normalized incident light into weapon position data.

The IST allows the WTET to sequentially track the X and Y weapon aiming position coordinates for up to nine trainees at roughly 30 Hz, the same rate a typical CCD based tracker would require to track one weapon. Due to the high speed of the IST, continuous data is available in real time for all active weapons. This data is subsequently used to determine real time weapon tracking, hit and miss data, and during replay, tracing with color coded icons, how each trainee moved his weapon during the scenario. This data is also used to determine when and if he pulled the trigger, and if he hit, wounded, or missed the intended target.

Trainee Location and Aggressor Shoot Back System - The WTET training environment allows for an enormous amount of data collection and flexibility. Three important aspects of the data collection during a scenario involve the following: 1) the location of each trainee, 2) the identification of each trainee, and 3) whether or not the trainee has taken proper cover from his on-screen adversaries and from possible friendly fire from other team members.

The method used by the WTET for trainee location, detection, and identification consists of multiple eye-safe Infrared Emitting Diode (IRED) arrays strategically located throughout the training environment. Furthermore, each trainee wears a MILES type torso harness with photo detectors and an audible Sonalert alarm. As the scenario progresses each IRED array is turned on in sequence, during the active time slot allocated for each trainee, such that the entire WTET training environment is mapped out in a grid like fashion. In this manner the location, detection, and identification of each trainee within any predefined zone is collected in real time and transmitted via an RF data link to the system computer during the allocated active weapon time slot for each trainee. A statistical prediction algorithm, based on the past movements of the trainee, is used to predict the current location of the trainee during the brief periods he is not detected by the torso harness photo diodes. The trainee avoids hostile fire (detection) as he would in the real world; he must take appropriate cover

as he engages the on-screen adversaries or risk being wounded or killed and thereby eliminated from the scenario.

The real time location and detection data of each trainee is used by the system computer to intelligently control the video branching (selection of sequential scenes to be displayed) as well as determine if and when the trainee is exposed to hostile fire from the on-screen adversaries. The location data in conjunction with the identification data is also used by the system computer to control the rifle sound effects. Statistical methods based on detection data are used by the system computer to determine the probability of a hit, miss, or wound. The location, detection, and identification data can ultimately be utilized during the replay for instructor analysis and debriefing.

Wireless RF Communications System - The wireless RF Data Communications System (WDCS) allows up to nine trainees to freely maneuver inside the training environment without being physically tethered or restricted to the system computer. This allows the trainees to evade hostile fire from on-screen adversaries as they would in the real world while moving to and from multiple rooms in a tactical manner. The WDCS further allows for the essential and rapid exchange of data between the system computer and the trainees for effective training.

The transceivers used in the Base Controller (BC) and the Weapon Controller (WC) are modular Spread Spectrum (SS) radio transceivers with no on board intelligence. These radios are controlled via a custom microcontroller interface for each particular application. By externally controlling the timing and the data protocol through embedded software the SS radio operation can easily be optimized for any one particular application. Utilizing spread spectrum technology also takes advantages of the high noise immunity, high data rates, and fast switching times associated with this license free technology.

The complete Spread Spectrum RF transceiver system consists of a central base station (Base Controller or BC) with 9 individual trainee located transceiver boards (Weapon Controller) for communication to and from the central base station. The trainee located transceiver boards are mounted on the torso harness and the batteries are located in the ammunition pouch on the trainee's side.

In operation, the BC receives an interrupt from the system computer and transmits a 10 bit data packet encoded with a unique start code, weapon ID, and sonalert status to all receiving weapons. Only one WC will recognize this data packet as being valid. All other weapons will continue to listen for a valid data packet while ignoring the current data packet. After the BC completes its transmission the BC is placed in the receive mode and waits for weapon status data from the active weapon.

The activated WC, having acknowledged the valid data packet, will proceed to control the weapon in question. The WC electronics "enables" the weapon mounted high-power eye safe IRED for the IST for approximately 3 msec. The WC is then placed in the transmit mode in preparation for transmitting the most recently acquired weapon status data. Once this data has been collected and stored in memory the WC transmits a 10 bit data packet encoded with a unique start code, detection data, low battery data, trigger data, selector data, and magazine data. The WC is then placed back into the receive mode waiting for the next valid data packet.

The BC decodes and acknowledges the data reception from the valid weapon and places the BC back into the transmit mode in preparation for the next system computer interrupt and subsequent weapon selection. The current data is then made available to the system computer. An error detection scheme, using the unique start codes and a watchdog timer, virtually eliminates the possibility of an erroneous response to an invalid data packet due to an RF data error.

Digital Sound System - The digital sound system provides a multitude of sound effects from various sources including commercially

available sound effects, actual live recordings from the field and a variety of synthesized sound effects. The sound effects, in conjunction with the video display, help to create a realistic atmosphere in which the trainee feels he is actually immersed within the training environment.

The digital sound system, in its current configuration, consists of a MIDI controller board, a sequencer, a digital sampler, a mixer, four processors, six audio amplifiers, eight speakers, and two sub woofers [2].

During an actual scenario the computer sends the appropriate commands to the sampler via the MIDI merge unit and the sequencer. The sampler recreates the appropriate sound effects from digitized samples stored in memory and sends the analog signals to the appropriate amplifiers through a mixer and four signal processors which drive foreground and background sounds.

SCENARIO INTERACTION

An expert system capability is built into the WTET to control scenario progression based on the overall training objectives and the behavior of the training team. A WTET scenario is composed of multiple video segments stored on video disc. Video branching allows aggressor targets to interact with the training team. Branching is accomplished by rapidly selecting and displaying video segments as the scenario progresses. Among the trainee behaviors which may affect branching are:

- Trainee / Team position
- Trainee/Team wound/kill status
- Weapon status (aim point, ammunition, shot fired, target coverage)

SCENARIO REPLAY / FEEDBACK

After a WTET mission is complete, detailed feedback of trainee performance is available through instructor station control. A variable speed mission replay using graphical icons and messages indicates trainees performance for the duration of the mission.

A hand held pointer (collimated IR source) connected to a trainee MILES type harness allows the instructor to control the replay. The pointer is controlled and monitored in the same manner as each trainee's weapon. A graphical cursor is displayed during replay to indicate the on screen position of the pointer. A graphical replay control menu is displayed during the mission replay. This allows the instructor to move through the training environment and stop, start, reverse and vary the speed of the synchronized mission replay.

During replay, the IOS synchronizes each Video Station's mission replay. Time into the mission is displayed as an event timer. Each trainee's continuous weapon aim point is indicated using graphical icons overlaid on a video replay. The icons are numbered to show trainee identification. Color coding provides weapon and trainee status. The color of the icon changes to indicate the following:

- Shot fired (miss)
- Shot fired (hit)
- Shot fired (wound)
- Shot fired (target kill)
- Trainee Hit
- Trainee Disabled

These graphical icons, along with event messages and a team performance summary, provide a detailed feedback of individual and team performance. In addition, a video recording of the training team's movements is displayed in synchronization with the after action review.

TRAINING EFFECTIVENESS

A training effectiveness test plan has been developed to test the WTET's individual device features and overall system effectiveness. This plan includes collecting subject matter expert ratings of device features as well as an attempt to establish student performance improvement as a result of using the system. Several local and

national law enforcement, as well as Department of Defense, agencies have expressed interest in evaluating this system. Data are currently being collect to determine what impact the major device features will have on improving trainee and team skills. These data will be presented at the Conference.

SUMMARY

The technological developments employed in the WTET improve training system fidelity for a variety of weapon team tactical training situations. The modular aspects of this system allow a great deal of flexibility in designing training environments and developing unique engagement scenarios. By accurately and continuously measuring trainee movements, weapon conditions and other interactive behaviors, instructors can provide detailed feedback on individual and team performance. The WTET prototype is ideally structured for agencies requiring close combat tactical training for individuals and small squads. The capability of the WTET to deliver a variety of threat scenarios is essentially limited only by what has been scripted and recorded.

REFERENCES

1. Marshall,. Wolff, McCormack, and Purvis. "Weapons Team Engagement Simulator", Proc.of the 12th Annual I/ITSC, Nov. 1990.

2. NTSC, "Weapons Team Engagement Trainer", Government Furnished Information, Dec. 1992.

SOFTWARE REUSE: A COMPANY VISION

Paul E. McMahon
Staff Scientist, CAE-Link Corporation, Binghamton, New York
Adjunct Faculty, Computer Science Dept., Binghamton University

ABSTRACT

Today, many exciting initiatives are underway within the software industry. Structure Modelling technology is growing rapidly through efforts at the Software Engineering Institute (SEI) and ongoing projects. Megaprogramming challenges are being faced on the ARPA STARS project. Open standards including POSIX, X-Windows, and Motif are becoming realities as key software vendors position themselves to support these initiatives.

Reuse library tools and guidelines are also being developed through efforts at the SEI, the Software Productivity Consortium (SPC), and on the STARS project. At the same time, software contractors are moving forward with serious strategies to improve their company software processes in response to industry initiatives including the ISO 9000 requirements and the SEI Process Maturity Model.

All these initiatives share the common objective of cost reduction and most are looking to one form or another of software reuse to achieve this goal.

This paper examines the multi-faceted issues of reuse and the role these current industry initiatives play within reuse technology. Issues discussed include analyzing existing software for reuse, techniques to design for reuse, reusable software architectures, managing variant versions of software, and managing a corporate reuse library. Technical and management issues are presented.

The paper focuses on lessons learned from efforts at CAE-Link to infuse software reuse techniques into the corporate culture. Practical techniques being applied today to meet reuse challenges are discussed. The key roles of reuse criteria, metrics, company software standardization, project-company interaction, management mandates and training and education are discussed.

Experiences and examples are provided from the B-2 ATD project, Independent Research and Development, and a corporate software Process Action Team that was instrumental in providing the focus necessary to move the company forward with an effective and practical reuse initiative.

ABOUT THE AUTHOR

Paul E. McMahon is a Staff Scientist at the Binghamton Operations of CAE-Link Corporation and an Adjunct Faculty member of the Computer Science Department at Binghamton University. Mr. McMahon has been with Link since 1973, holding various technical and management positions within the company. Mr. McMahon has published numerous papers on Ada and software engineering including a paper entitled "Lessons Learned on the Fringe of Ada", which was nominated for best paper at the 1989 Interservice/Industry Training Systems Conference and a paper entitled "Software Metrics, Ada, and the B-2 ATD", which was awarded best paper at the 1991 Interservice/Industry Training Systems and Education Conference. Mr. McMahon teaches Software Engineering at Binghamton University.

SOFTWARE REUSE: A COMPANY VISION

Paul E. McMahon
Staff Scientist, CAE-Link Corporation, Binghamton, New York
Adjunct Faculty, Computer Science Dept., Binghamton University

INTRODUCTION

What is Software Reuse?

To many, software reuse means code. Reuse, however, is multi-faceted. In fact, its greatest potential for cost reduction may be found in other software forms such as requirements, design, documentation, and the development process itself. Most software industry initiatives today are striving for one or more of these forms of reuse. See Table 1.

Form of Reuse	Industry Initiative
Requirements	COA, CASE Tools, Workstations
Design and Documentation	OOD, Structure Modelling, Megaprogramming, CASE Tools, Workstations, Reuse Library
Code	OOP, Variants, Reuse Library, Autogeneration
Test Cases	CASE Tools, Regression Testing
Development Process	SEI Maturity Model, ISO 9000

Table 1 Forms of Reuse

Tools and Techniques

Object Oriented tools and techniques are popular today largely due to their potential to provide more reusable products. CASE tools provide the potential for standard representations resulting in reusable requirements, designs and documentation. Reuse tools can aid organizations as they transition to software processes with more of a focus on reuse.

Software Architecture

Structure Modelling and Megaprogramming initiatives each provide forms of software architecture reuse. Structure Modelling focuses on reuse through design commonality. Once common design elements are identified, reuse can be enforced through the use of structure model templates.

Megaprogramming focuses on reusing complete software infrastructures. "Megamodules are independently maintained software systems managed by a community with its own terminology, goals, knowledge, and programming traditions."[1]

Distributed Interactive Simulation (DIS) is a form of Megaprogramming that reuses complete training devices to face new training needs despite the fact that these devices were never envisioned to operate in this fashion. Communication and control through standardized protocols are key attributes within this emerging technology.

Open Standards and Process Improvement

Efforts geared toward industry open standards and company process improvements provide other forms of reuse.

Open standards means software is reusable across vendor platforms.

Company process improvement efforts provide standardization and repeatability of software processes across company projects. This results in reuse of procedures, tools, training and even corporate knowledge.

Software Development efforts of the past relied heavily on the knowledge and opinion of subject matter experts to keep projects on course. Today, company initiatives are moving away from reliance on individuals toward defined and managed processes that are repeatable, cost-effective, and independent of the skills of particular individuals.

Why is Software Reuse Important?

The primary reason software reuse is key today is the need to improve cost-effectiveness. Finding better ways to reuse software to meet our needs means that we can do more for less. This translates into

reduced development cost, reduced risk and increased reliability and maintainability.

REUSE TECHNIQUES

Due to the many forms of reuse, there are also a number of implementation techniques.

We have identified six key reuse techniques at Link.

1. Analyzing Existing Software
2. Designing For Reuse
3. Structure Modelling
4. Variants
5. The Corporate Reuse Library
6. Management Support and Mandate

Analyzing Existing Software

The fact that software exists does not imply its suitability for reuse. Existing software must be analyzed against established criteria to determine its reuse potential. This analysis includes answering the following key questions:

1. What do we want to reuse (design, documentation, test cases, code)?
2. Is the design compatible with the planned software architecture?
3. Do we want to reuse only the algorithmic design or more?

It is currently believed that the most valuable reusable software products may be analysis and design. In many cases, math models may be reusable, but coded modules may not be compatible with modern software architectures. Reuse analysis is necessary to answer these questions.

Designing For Reuse

Designing software to be reusable may cost more. This is because our development model and techniques change when we are designing for reuse. This is an investment that will begin to pay off as the reuse library is populated. The reuse library is discussed later in the paper.

We have identified four key factors in designing reusable software. It is recommended that these factors be employed as a review criteria for determining acceptability of software for the reuse library.

1. Reuse Existing Software First

When designing reusable software, our development model changes. The first step is to look to reusable components to meet needs. Our goal is to build solutions from existing library components rather than reinvent new solutions. Technical trade-offs may be necessary.

This may mean early discussions with the customer refining requirements to support maximum reuse to reduce overall cost. Customers will need to be aware of these changes in contractor reuse processes.

2. Follow Standards

Follow company software standards to ensure new software meets reuse criteria. In a reuse development environment key standards include software naming conventions, standard software packages, and established architecture guidelines.

3. Isolate Device Specifics

Device specific data must be isolated minimizing the effort required to adapt reusable software to changing requirements. The management of modified reusable software is discussed further in the section on variants.

4. Use Object Oriented Techniques

Current studies and our experience to date indicate object oriented techniques result in more reusable and maintainable software than traditional methods.

Structure Modeling

Work at the SEI and experiences on the B-2 ATD indicate that structure modeling techniques are effective at enforcing reusable common designs, simplifying training, and reducing schedule and computational resource risks.

Engineering productivity may also be enhanced through autogeneration of structure model components. CASE tools may be used to enforce common design decisions supporting the structure model.

Variants

The Variant aids in the management of reuse during software modifications. A variant is a software component with a special relationship to a "parent" software component. Both the parent and the variant

are independently managed and tracked, but the variant reuses a significant amount of the parent software.

There are similarities between the concept of variant and the concept of inheritance found within object oriented languages. Inheritance provides a form of reuse, but may include a run-time penalty.

Variants, on the other hand, are managed through an off-line configuration management system eliminating run-time overhead. The off-line system manages the relationships between "parent" units and variant offspring tracking and measuring variant reuse.

Variants provide a powerful mechanism to manage large collections of reusable software components across many similar, but functionally modified systems.

Corporate Reuse Library

A corporate reuse library is critical to the success of a company-wide reuse effort. However, equally important to its functional capabilities, the library processes and procedures must be integrated with company existing software processes and tools. This includes approvals, reviews and change notifications.

While many large companies do not yet have reuse technology as part of their software process model, they may have well-established software configuration management tools and procedures. The infusion of reuse technology into the company must minimize redundant engineering effort. In particular, reuse processes and tools should be integrated closely with existing software configuration management processes, ensuring that identification, statusing, approvals, and notifications are processed as efficiently as possible.

Management Support and Mandate

Changing a company's software process model to effectively support reuse requires more than a reuse library and a technical understanding of the issues. Software engineers frequently prefer to reinvent rather than reuse unless clear direction to do otherwise is provided.

For reuse technology to effectively take hold in a corporation, it is essential to educate all levels of software management in the latest reuse principles and strongly support the company reuse effort. A management mandate to follow the principles of reuse must be clear to all involved in software production. To manage this effort successfully, reuse objectives should be established. A reuse program should include metrics and feedback of results, taking corrective action where necessary. Through the reuse library, integrated closely with a disciplined configuration management system and strong management backing, the capabilities exist to measure, manage, and succeed with reuse technology.

Past attempts at Corporate level reuse have failed largely for four reasons. First, criteria, control, and approval were unclear. This resulted in product changes initiated at the project level that were inconsistent with the company vision. Second, adequate resources were not supplied to support the company perspective. Third, motivation from the organization continued to be the project rather than the company. Fourth, the notion of reuse as code only was a barrier. Change in each of these areas is essential to the success of a corporate reuse effort.

CAE-LINK INITIATIVES BACKGROUND

In the first half of 1992, an outside consulting organization was placed under contract by CAE-Link to facilitate Company-Wide process improvements. A software process action team (PAT) consisting of CAE-Link senior engineers and software consultants was formed. The objective of the team was to study current software policies and processes at Link and implement improvements necessary to reduce software life-cycle cost.

One and one-half years earlier an effort was initiated to export the B-2 ATD Ada software process and tool-set, making it available for other CAE-Link projects. Since that time, enhancements to both the tools and the software process have continued on the B-2 project and through Independent Research and Development at Link. The information presented in this section is based on these activities.

Software Process Action Team (PAT) Lessons Learned

The Software Process Action Team (PAT) conducted interviews throughout the

Company with both junior and senior software engineers and software managers to identify key areas for potential improvement. Seven target areas for improvement were identified as a result of this activity. These include:

1. Eliminate process redundancies and non-value added work.
2. Do not release (put under formal configuration control) software until fully tested.
3. Improve the off-line test environment and tools.
4. Focus on early error detection.
5. Improve software training.
6. Establish company level standard metrics.
7. Establish a Corporate software reuse library with defined procedures for reviews, approvals, and reuse criteria.

Process Action Team Results

In parallel with the interviews, the PAT members and the consultants met periodically to brainstorm potential improvement strategies. Presentations from experienced senior technical engineers frequently provided the focus for discussions.

As a result, action plans were established and carried out. The company Software Standards and Procedures Manual was modified addressing targeted areas for improvement.

Formal company software training classes were prepared and conducted in support of these initiatives. This training included:

1. Object Oriented Techniques
2. Criteria for:
   a. Releasing software
   b. Design level of detail
   c. Risk assessment
3. Standard company design representation
4. Standard company Structure Model
5. Standard company software metrics
6. Standard company software process
7. Software management techniques (i.e., algorithmic cost estimating).

See Figure 1.

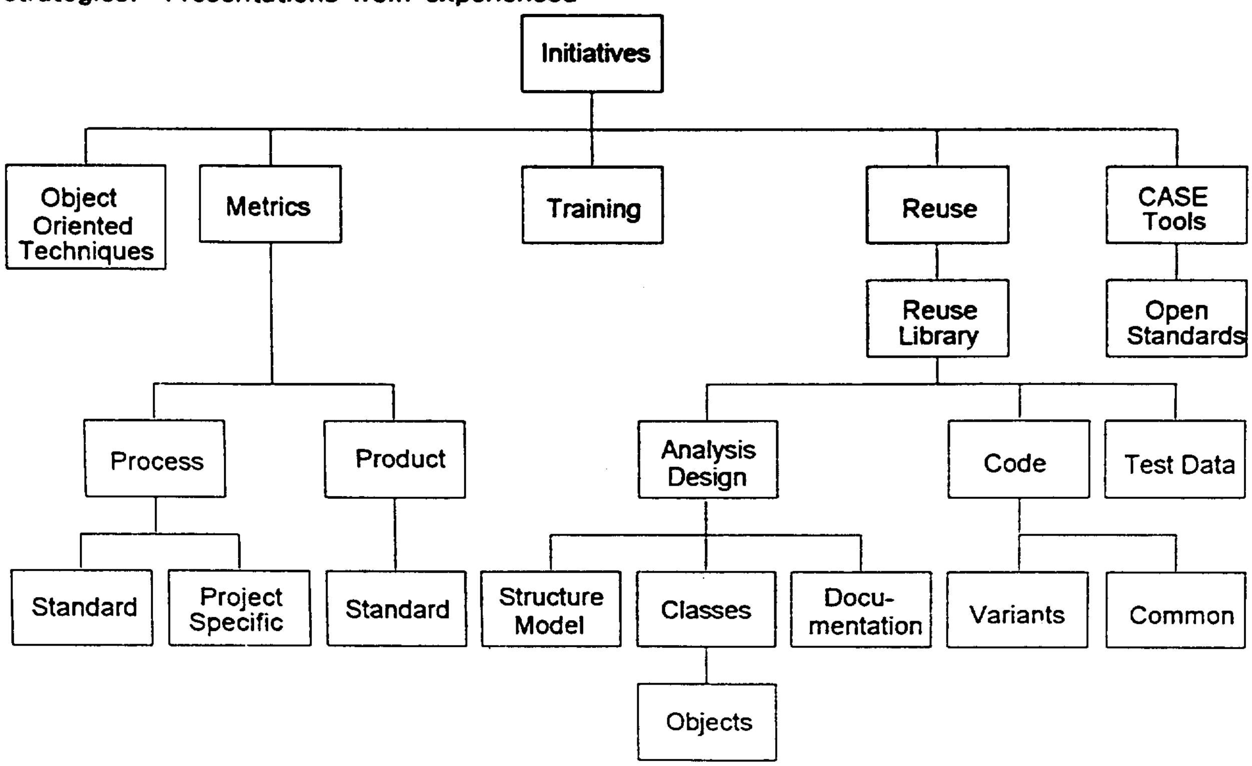


Figure 1 Process Improvement Initiatives

Reuse Through Process Improvement

Making the process repeatable is another form of software reuse. The SEI Capability Maturity Model defines five process levels. See Table 2.

Level	Capability	Description
1	Ad Hoc	Process not repeatable
2	Repeatable	Repeatable, but dependent on people
3	Defined	Procedures and Process Training
4	Managed	Metrics collected
5	Self-Improving	Metrics feedback for improvement

Table 2 SEI Process Maturity Model

Our objective is to reuse training, tools, procedures, and standards company wide. However, it is critical that any methods reused company wide be as cost-effective as possible. This implies that the company standard method must be self-improving. Repeatable and defined is the first step. However, a self-improving process must be our ultimate objective (SEI Level 5).

To achieve level five, metrics are necessary. Selecting the "right" set of standard metrics for your organization is a key to success. The right set for one organization may not be right for another. The metrics chosen must add real value to each organization's own process, while, at the same time, adding minimum burden to the software engineer.

Our software processes at Link are based on a combination of modern software engineering principles, and experiences and lessons learned from real-world projects. Metrics provide an example of this blend of real-world experience and textbook theory.

METRICS

To ensure our reusable processes, procedures, and tools are as efficient as possible we must measure. In establishing our company approach to metrics, we listened to the people from our ongoing projects.

We found from our experience in applying metrics on the B-2 ATD project that there are two distinct kinds of metrics necessary to support real process improvement. We refer to these two types as Standard and Non-Standard metrics.

Standard Metrics History

Standard metrics are those that are applicable to all software developed at Link. Standard metrics are collected periodically and are used as feedback for process improvement.

As part of our PAT interviews, we asked managers and engineers for feedback on the effectiveness of standard metrics. These interviews provided us with some key insights.

First, we found that standard metrics were not being collected consistently across all projects. Second, we found that the lower one moved into the organization, the less value was reported with the use of these metrics. Metrics were not being collected consistently because the engineers and their immediate supervisors found minimal value in this data.

Project engineers, however, reported that software metric reports were found to be useful. Project engineers used metrics to identify trends and potential problem areas early.

Standard Metrics Lesson Learned

Engineers and first level managers tend to be close to the problems on a daily basis. As a result, the value of metrics as a management aid at this level is low. Metrics are trend indicators. The higher one's perspective or span of accountability, the more valuable they become.

We found that metrics are particularly valuable to those concerned with multiple projects. Metrics can provide a common ground for comparison leading to better understanding of the root cause of problems.

The Process Action Team established a company set of standard software metrics, and initiated training of all personnel including engineers and managers in why it is important to collect this data accurately from a company perspective

We found that people respond positively to initiatives when they comprehend the

motivation. This means education. Metrics is becoming an integral part of our training program as well as our company software vision

CAE-Link Standard Metrics

1. Cost
2. Schedule
3. Manpower
4. Execution Time
5. Memory
6. Complexity
7. Source Lines of Code (SLOCS)
8. Stability (Rate of Change)
9. Problem Category.

Non-Standard Metrics

The value of metrics rests in process improvement. Optimum process improvements can only take place if the "right" information is available to detect process weaknesses This may at times require collection of "non-standard" metrics.

Non-Standard Metrics can include any data necessary to understand a perceived problem. They can include such measures as the length of time for an engineer to complete or learn a sequence of process steps perceived to be inefficient. Non-Standard metrics are only gathered for the period of time necessary to isolate a problem, and implement a solution. Once resolved, for efficiency, collecting of non-standard metrics should cease.

On the B-2 ATD, in response to a problem report on the build process efficiency, the following non-standard metrics were collected over a period of several months and analyzed weekly:

1. Elapsed wall clock time of each load
2. Number of changed software units per load
3. Number of lines compiled
4. Number of tasks linked
5. Number of load build process problems
6. Categories of build process problems
7. Elapsed time of segments of build process.

As a result of this analysis, a process improvement plan was initiated. Lessons, rules, and guidelines resulting from this activity were communicated to the company Process Action Team for approval and incorporation into the company standard process. It was the collection of specific non-standard metrics on a project that provided the needed insight leading to key process improvements for the company.

SOFTWARE CONFIGURATION MANAGEMENT

Earlier in this paper key areas targeted by the Company Process Action Team were identified. One area identified was Configuration Management.

Interviews with engineers indicated that certain projects may have applied too much control too soon resulting in unnecessary process inefficiency. As a result, the Process Action Team collected data, examining a sampling of systems across multiple programs. We found that certain systems had been placed under configuration control prematurely and some programs were requiring too high a level of change approval too early. Releasing software before it was adequately tested was found to be the result of inadequate education and training concerning the relative cost impact of detecting errors prior to (versus after) release.

The Cost To Detect and Fix Errors

The cost to detect and fix errors during integration is significantly greater than the cost to detect these errors prior to release.

When design errors are not detected until integration, the impact is great for the following reasons:

1. Load build time is extended. This affects many engineers.
2. The rigor of the change process (approvals, etc.) is more costly.
3. Having software in the load that has not been fully tested can cause testing rework to interfacing systems.
4. Possibly the most significant, frustration caused by all of the above leads to low engineering morale.

Company Action Plan

The Process Action Team determined that the root cause of reported inefficiencies

was not the company software configuration management policy or procedures, but rather a misapplication of the process and inadequate training and education in the consequences of premature release and excessive early controls. This education became a key part of our formal software engineering training program.

It was the key feedback from metrics that initiated the actions leading to these key improvements in our process.

PROJECT AND CORPORATE INTERACTION

Project feedback to a company focal point is critical to effective company level process improvement and reuse.

As a result, a corporate level Software Engineering Process Group (SEPG) was permanently established at Link. The SEPG listens to project concerns and lessons, approving, where appropriate, Company software process changes. Any changes to the Company standard software process and supporting tool-set must be approved by the SEPG. See Figure 2.

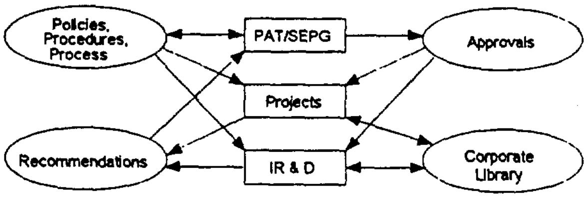


Figure 2 Project and Corporate Interaction

SEPG decisions are based on the company software vision. Software process changes or tool modifications are approved only if the change is in the best interest and consistent with the Company software principles and vision.

In the past, individual projects have modified software tools and processes based on shortsighted project issues. This resulted in overall increased software development and training costs.

Each directorate within CAE-Link with software responsibility has a representative on the company SEPG. It is through this vehicle that we are affecting the changes necessary to make reuse technology integral to our software development process.

WHERE ARE WE GOING?

Today we are recognizing the needs of the full life-cycle of software. We are using front-end analysis and design tools and object oriented techniques to better communicate with our customers and ourselves. These techniques and tools also provide more reusable software.

Reuse techniques, methods, and tools are certain to play a key role in reducing future software costs, allowing us to do more for less. We envision a reuse library integrated with configuration management systems providing disciplined and efficient software processes supported by metrics and feedback leading to continual process improvement.

We are positioning ourselves to grow through open standards and commercial off-the-shelf software products. We are moving toward improved off-line test environments supporting earlier less costly error detection.

Today, all the tools required to support our vision are not yet commercially available. We plan to support our company process with the necessary tools developed and managed through the corporate library and the SEPG.

As improved products become commercially available we intend to maximize our use of commercial products supporting open standards.

BALANCING STANDARDIZATION AND GROWTH

Companies that fail to change will not survive. At the same time, too much change or change of the wrong kind can be equally detrimental to the success of a software organization.

Each company must establish its own vision of the future with clear software objectives and change guided by its own objectives.

CONCLUSIONS

Early reuse programs will not be capable of addressing all associated issues. Companies seeking to institute reuse programs must look closely at their own process and products to determine where the greatest gains are to be made and how to best integrate reuse technology into current software cultures.

Successful reuse initiatives today demand a selective and focused strategy coupled with management mandates, training, and education.

Software reuse is not limited to our products. Perhaps the greatest potential for cost savings is found in the reuse of efficient processes. The path to better processes and methods is through metrics.

Without measurement, we tend to work bottom-up with decisions being based on nearsighted perceptions. It is frequently these decisions that produce the products that are difficult to reuse and maintain.

Effective reuse programs must be integrated with process improvement continually providing feedback through metrics. Process metrics should be viewed from the company perspective. We need to change to survive, but change must be consistent with our principles and our vision.

REFERENCES

1. Wiederhold, Gio, & Wegner, Peter & Ceri, Stefano (Nov. 1992). Toward Megaprogramming. Communications of the ACM.

APPLYING ADVANCED PARALLEL PROCESSING CONCEPTS TO RADAR SIMULATION AND IMAGE GENERATION

Edward W. Drew and Ron Matusof
CAE-Link Corporation
Binghamton, New York

ABSTRACT

This paper discusses advanced parallel processing concepts and their use for radar simulation and image processing. It describes both the advantages and disadvantages of a number of architectures and illustrates these with actual implementations. It discusses issues relevant to real-time image generation, including latency, synchronization, and scheduling dispersion. It also discusses the problems inherent in designing state-of-the-art systems in a research and development environment, and then applying that product to an evolving market. Finally, it makes recommendations concerning future directions in parallel processing and simulation.

ABOUT THE AUTHORS

Edward W. Drew is a senior staff engineer with the CAE-Link Corporation, Binghamton, New York, and has 25 years of experience in tactical and radar simulation. He is currently assigned as systems engineer for the pDRLMS product line, with responsibility for radar simulation for several flight training devices. Mr. Drew holds a MS in Physics from the State University of New York at Binghamton, a D.C.Ae in Aviation Electronics from the Cranfield Institute of Technology, England, and a B.Sc in Physics from the University of Bristol, England.

Ron Matusof is a staff engineer with CAE-Link Corporation, Binghamton, New York. He has over ten years of experience in tactical, radar, and image simulation and simulator networking. He is currently acting as Principal Investigator for CAE-Link's Transputer Modular Processing System. Mr., Matusof holds a Bachelor of Science in Electrical Engineering from the University of Pittsburgh. He has published several papers on the subjects of Interoperability, mission rehearsal, and cue correlation.

APPLYING ADVANCED PARALLEL PROCESSING CONCEPTS TO RADAR SIMULATION AND IMAGE GENERATION

Edward W. Drew and Ron Matusof
CAE-Link Corporation
Binghamton, New York

INTRODUCTION

Throughout the history of simulation, one of the most complex problems has involved the synthesis and generation of imagery that represents the simulated terrain. This imagery, whether it is for out-the-window viewing, simulation of infrared (IR) for night vision operations, or simulation of radar imagery, has usually required large processing capability and expensive, custom designed hardware and software.

In the last few years, three conflicting forces have radically changed the approach to the simulation of imagery. First, real-world equipment (such as IR sensors and radars) has become significantly more complex. This has tended to drive up the complexity of image simulations. At the same time, competitive pressure has forced a decline in the price of image simulations by at least one order of magnitude, and we can expect this trend to continue in the future. Finally, as defense budgets continue to decline globally, systems originally designed for military use are being converted into more commercial applications, which tends to both add competitive cost pressure and at the same time increase performance requirements. The net result is that the simulation of imagery during the mid 1990's will have to perform more detailed computations across a wider number of applications, and at a significantly lower cost than systems designed just ten years ago.

In 1988, we began work on a data flow architecture for multi-mode radar simulation.[1] This work was performed for four years under research and development funding with the intended goal of producing a radar simulation system with increased processing capability and an order of magnitude reduction in recurring cost.

We were interested in applying parallel processing techniques to lower the cost and increase the performance of our simulation. Parallel processing allows for increased computational performance at significantly lower recurring costs, but carries with it a set of unique design methodologies and constraints. The architecture we describe in this paper has been applied to a variety of Navy, Air Force, and International Digital Radar Land Mass Simulation (DRLMS) programs. Additionally, the same architecture (and, in fact, the same hardware design) supports US Army helicopter combat training by providing environmental feedback information.

In this paper, we discuss our approach to parallel processing, including both the advantages and disadvantages we have discovered after six years of product development, testing, and fielding.

PROBLEM DEFINITION

The basic ideas behind most radar simulations involve simulating the signal at various stages of its life, including: its emission by the radar, its propagation, effects from reflection off terrain and cultural surfaces, the signal's return propagation, and the radar internal signal processing characteristics. The degree to which the signal is simulated at each stage determines the overall fidelity of the radar simulation.

High fidelity radar simulation is an extremely complex undertaking. The equations that govern the radar signal propagation (namely Maxwell's equations) are not well suited to operate in discrete time steps, like those found in most simulations. On the other hand, the effects of the signal propagation are either totally independent (as when terrain is illuminated) or additive (for example, when the received signal is processed). The independent and additive nature of radar makes it a primary candidate for parallel processing.

Similarly, the problems faced by the designer of image generators are highly complex and require large processing capabilities (many estimates range from one billion to one trillion floating point operations per second). One subset of the image generator is the function of environmental feedback. Environmental feedback refers to the

processing that provides information concerning the interactions between any two points in the simulated environment (such as line of sight, collision detection, surface attributes, etc.). These interactions are usually independent and we find that they are ideally suited to parallel processing.

Our intent was develop a parallel processing architecture that could apply to a wide variety of complex applications including radar simulation, image generation, acoustic simulation, acoustic analysis, and the like. Although we have been generally successful in our system development, we have learned some interesting lessons concerning parallel processing, radar simulation, data base manipulation, and product development using state-of-the-art techniques and hardware.

PARALLEL PROCESSING

Conventional computer processing is a sequential task where program execution occurs in a pre-defined order and operates under the control of a central processor. Although the speed of conventional computer processors continues to improve dramatically, the highest processing throughput attainable is still limited by the speed of the central processor, its ability to access memory, and the ability to move data between the processor and input/output ports.

Recent enhancements to serial processors have included a super-scalar architecture, which, in theory, allows several instructions to be executed simultaneously. In practice, there are only particular combinations of instructions that execute simultaneously, such as operand and instruction fetch, so performance improvements vary greatly between applications.

Parallel processing uses a different approach to improve the processing power of a computational architecture. Rather than using the conventional von Neumann model of a computational machine, parallel processing divides the problem into a number of individual tasks that are processed independently. In many applications, the problem is divided across multiple processors where the individual processor is still a von Nuemann machine operating in serial fashion. Although this approach provides significantly greater overall processing power, it is still constrained by the bandwidth of the interconnection between the individual processors.

Another approach to parallel processing divides the tasks into a number of software processes and interconnects these processes through a data flow path known as a link. This approach separates the processing architecture (a software function) from the physical architecture (a hardware function). The software design does not constrain itself to a von Nuemann architecture (even if housed on von Nuemann computational platforms) and significant processing improvements are theoretically possible. The overall system performance, however, is limited by the choice of computational processors, the mapping of software processes and links to these processors, and the physical interconnection of the processors.

ADVANCED CONCEPTS

After a great deal of research, we chose to implement a prototype programmable DRLMS (pDRLMS) on an interconnected network of Inmos (now SGS-Thompson) T-800 transputers. Transputers are powerful 32 bit processors that support high level language development, parallel on-chip processing, and include four high speed data link connections to other transputers. In 1988, when we started this program, these processors were among the fastest processors on the market. They were also relatively new to the market and very few applications had been implemented on them.

The pDRLMS uses a data-flow architecture and its design attempts to separate the software implementation from the hardware architecture. Data flow architectures are those in which the messages that flow between nodes provide control and synchronization of the system. Software tasks are divided into processes that communicate with each other via virtual links. A virtual link does not necessarily have a corresponding physical link, and the mapping of processes (software tasks) to processors (hardware nodes) is usually not one-to-one.

Traditional DRLMS applications involve some form of pipeline, where each process acts as a worker on an assembly line. Data comes as input from the previous worker, processed, and then output to the next worker. Many attempts to design parallel

DRLMS implementations have involved making parallel pipelines, so that numerous processes occur concurrently. Unfortunately, the start of the pipeline (data transfer from the host computer) and the end of the pipeline (display of imagery to the crew) can not be broken into multiple parallel tasks, and these become bottlenecks in the system.

A different problem occurs for applications that involve large data base processing capabilities, such as environmental feedback calculations. In these cases, the problem can be decomposed until there is a one-to-one mapping between data base polygons and software processes. In this extreme case, the benefits of parallel implementations are lost to the overhead of communicating between a large number of processes.

The degree to which a problem space is made parallel is known as its granularity. Coarse grain parallelism occurs when the problem is broken into very large pieces. Fine grain parallelism occurs when the problem is broken into very small pieces (such as a small block of sequential code). When a problem is too coarsely parallel, throughput suffers since there is a large amount of sequential processing in each software process. When the problem is too finely parallel, system performance degrades due to an increase in message traffic between the processes. We therefore chose a methodology of decomposition that optimized the granularity for the most efficient processing architecture. This method is iterative, and attempts to decompose the problem in four ways:

1. Functional Decomposition. During functional decomposition, large functions that operate independently are identified. For example, the major functions in a radar simulation might be data base manipulation, radar illumination, radar effects, and radar image generation.

2. Domain Decomposition. During domain decomposition, the data which is to be processed is examined to determine if there are subsets (domains) which can be conveniently grouped to provide increased processing efficiency. The domains required by each major function are then identified. Domain decomposition is very useful in reducing the amount of processing a single node must perform.

3. Farming. A farm is an architectural concept where each processor performs the whole task on a portion of the domain. This scheme allows scaling of the processing. The processing power can be increased by increasing the size of the farm.

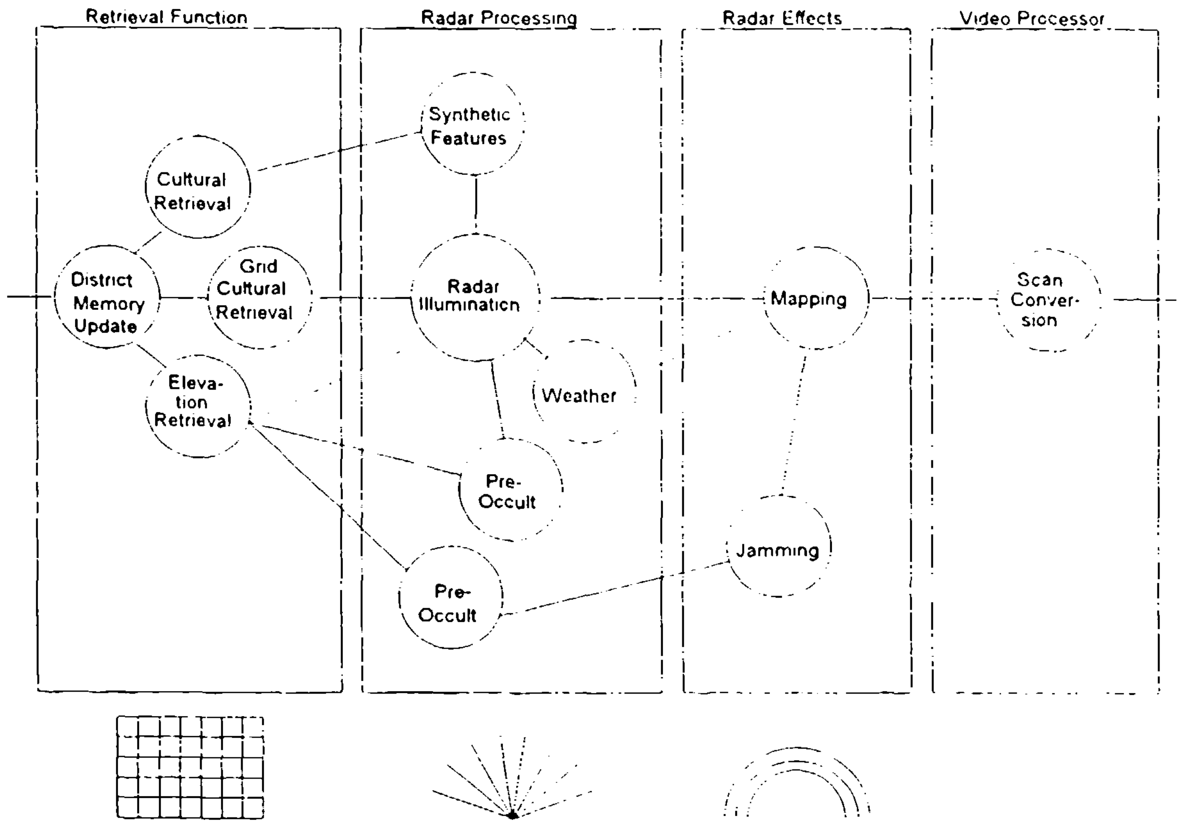
4. Pipelining. A pipeline is the antithesis of a farm in that each processor performs a portion of the task on the whole domain. In many applications, there is great efficiency gained by breaking a single serial task into a set of smaller tasks operating in a pipeline.

Once the problem has been fully decomposed, the required replication of tasks is determined. The appropriate number of times a process, farm, or pipeline is replicated is a function of the desired throughput of the system, the available resources, the desired traffic on the network, and the degree of parallelism inherent in the problem space. This represents a delicate balance between system performance, system cost, and system reliability and these decisions are better made on a case-by-case basis.

The decomposition of a simple DRLMS is shown in Figure 1. Although the decomposition in the previous paragraphs describes functional decomposition, the methodology described works equally well for other decompositions, such as object-based or data-driven decompositions.

Once the software processes have been identified, they are mapped into a software architecture known as a *virtual network*. The virtual network describes the interconnection of individual processes and the communication and data path between them. There is no requirement for the virtual network to map one-to-one with a *physical network*. It is because of this separation of virtual and physical networks that it is possible for a process to have large numbers of virtual connections, while the host processor has only four physical connections.

Our architecture is based upon a virtual intercommunication scheme which make



Grid Domain Operations provide efficiency for data base manipulations, such as retrieval.

Radial Domain provides efficient structures for radar processing, such as Radar Shadowing

Cross Range Domain provides efficient structures for radar effects such as Azimuth Beam Spread

Pixel Domain is required for video processing

M-4

Figure 1 Simple DRLMS Decomposition
Processing is divided into functions and domains.

the physical implementation of the network transparent to the application software. This capability is provided by a software package known as the message handler. The message handler resides on each physical node and it acts as a postman, sending messages to the correct destination, and receiving and buffering incoming messages. Messages are received via any of the physical links and buffered. Once messages are received, the message handler re-transmits those messages destined for other nodes (a process referred to as ``through-routing'') and organizes the messages intended for this node. Based on the incoming message traffic, the message handler ``wakes up'' processes which have turned themselves off while waiting for data and assigns them to execute a particular task. The message handler also transmits messages created on this node by resident processes.

APPLICATIONS

Our first application of this architecture was in support of the AH-64 Combat Mission Simulator for the US Army. The architecture was used to implement a Terrain Information System (TIS). The TIS is an environmental feedback system which provides high-fidelity line-of-sight, elevation, and occultation calculations which are fully correlated to the simulator's image generator data base.

The TIS is a classic example of domain decomposition and farming. All information passed between the host computer and the TIS travels through a single process known as an Interface Manager. This process determines what information is required and which part of the network will provide it. The Interface Manager then requests that environmental feedback information be provided by one of the four farms comprising the TIS. Each farm is composed of one Executor, which is responsible for coordinating the activity of the farm, and fifteen model processors. Each model processor has a unique domain, consisting of a subset of the data base. The Model Processor operates upon its domain, with the Executor correlating the results and sending the information back to the Interface Manager for distribution to the host. As such, the problem of environmental feedback has been domain decomposed at the Model Processor level and farmed at the Executor level. The TIS architecture is shown in Figure 2.

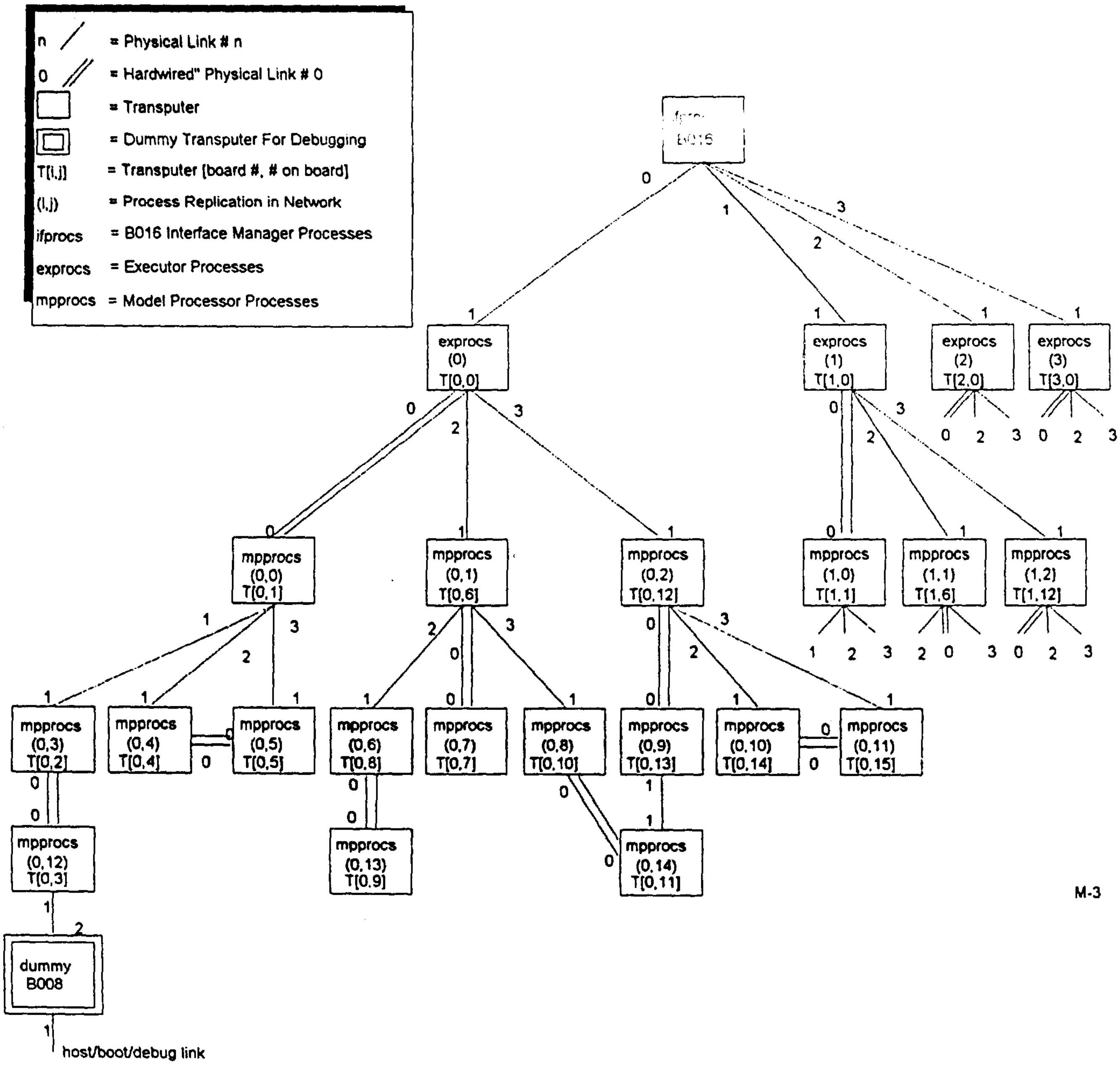
Our second venture was a production version of our research and development pDRLMS. The pDRLMS is an example of all four methods of problem decomposition. The DRLMS was first decomposed into major functional processes of data base retrieval, illumination, radar characteristics, and network control.

Several domains became apparent at the outset. Three types of data base are required for the pDRLMS, namely gridded terrain, list-based culture, and weather descriptors. Although these could be viewed as separate domains, we opted to consider them as a single domain consisting of information which is defined in spatial terms (i.e., absolute position and attitude). Radar effects, on the other hand, tend to happen in a series of sectors emanating radially from the radar transmitter and coincident with the sweep pattern. We defined this domain to consist of information which is defined in radial terms (rotation and range) from the transmitter. Beamspread effects are a function of range and have very little contribution from rotation. We opted to define a third domain for beamspread which is based solely on range from the transmitter. A final domain was reserved for video generation, which requires information to be described in terms of pixel space.

With our major functions defined and our domains identified, we proceeded to determine where farming and pipelining would be beneficial. The result is shown in Figure 3.

ISSUES AND LESSONS LEARNED

Our architecture is asynchronous, and it brings with it certain concerns inherent with the asynchronous generation and use of data. The system is synchronized to ``real-time'' in at least one place: the interface with the host computer. In the case of pDRLMS applications, it is also synchronized to the frame rate of the radar display. Between these points, the system is synchronized only by the flow of messages. It is therefore possible to envision situations where data consistency across the network is not achieved. We had feared in certain applications that it might be possible to generate erroneous data due to data inconsistency, but thorough testing of the system



M-3

Figure 2 TIS Physical Network

Each Executor Processor Coordinates the activities on an individual farm of model processors.

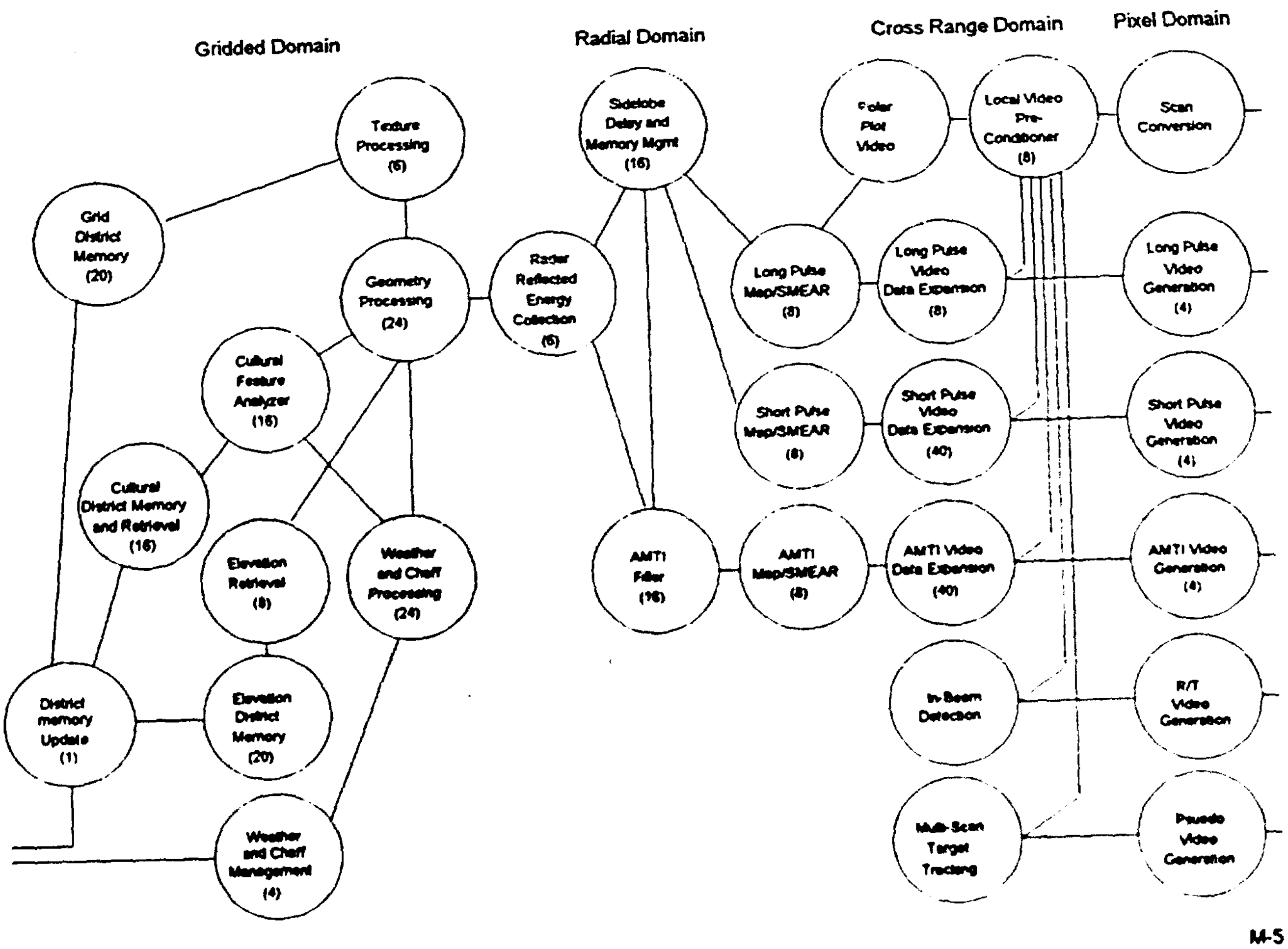


Figure 3 Simple DRLMS Decomposition

System has been decomposed into major functions and domains (see Figure 1) and is further decomposed into pipelines and farms. Numbers in parenthesis indicate the size of the processing farm (number of parallel processes).

by both CAE-Link engineers and our customers has failed to detect this problem.

Similarly, we had been worried that in very large networks, scheduling dispersion could potentially cause unsynchronized updates which would be perceivable by the crew, and again this has not been the case.

This is not to say that our experience with this development has been without problems. One of the first problems we encountered was the support for high level languages. Although compilers were available for both FORTRAN and C, the only truly supported compiler was provided for OCCAM. OCCAM was the first language to be based upon the concept of parallel and sequential execution, allowing for communication and synchronization between concurrent processes. Although the language is quite efficient and provides significant advantages for parallel processing, we have found that it is equally hard to find qualified OCCAM programmers. As the compilers for C and C++ have improved, and libraries have been added to support parallel execution, we have begun to reduce the amount of OCCAM in the system with the eventual goal of eliminating OCCAM entirely.

There were few applications hosted on T800 Transputers when we began this design effort in 1988. The size of our networks (between 64 and 400 physical nodes thus far) required special considerations in our design. We developed our message handling software largely because there was no commercially available alternative from which to choose. We therefore have invented a proprietary, closed architecture in a situation where we had desired a fully open architecture. As the use of transputers grows, commercial alternatives are becoming available and it is our goal to migrate to a commercial message handler system in the near future.

We also tended to find quirks in the prototype hardware with which we were working. To be fair, we were generally using pre-qualified hardware and beta releases of software tools and often found ourselves in the unenviable position of working around unanticipated hardware shortfalls.

Our method of decomposition appears to have worked quite well, although it still leaves a bottleneck at the interface to the host and at the output to a crew display. However, we have successfully reduced the cost and complexity of a typical radar application by over an order of magnitude and have implemented the TIS with only one board type and the pDRLMS with only four.

CONCLUSIONS

In general, we have found that our architecture, and most other parallel architectures, offer the following advantages over traditional approaches:

1. Scalability. Our system has been scaled to a factor of 12-to-1 increase in performance simply by adding processors, and we believe that a 32-to-1 increase is readily achievable.

2. Reconfigurability. Because the hardware, the communication software, and the application software are designed independently, a wide variety of applications can be hosted on this architecture.

3. Optimization. This architecture allows processes to be moved from processor to processor without affecting the hardware interface. This allows for easy load balancing, additional problem space decomposition even after the system has been built, and the ability to optimize by adding additional processes as needed.

We are firm believers in the concepts of parallel processing. There are many fine parallel processing architectures on the market, and we are not attempting to imply that this architecture is inherently better than any other. We believe that parallelism can be successfully exploited as a general purpose architecture to solve a variety of problems, and we propose our four-step method of problem decomposition as an attractive method of determining the appropriate level of granularity for a given problem space.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the contributions of the design engineers who created and developed the CAE-Link parallel processing architecture over its six-year life, and in particular, Curt Carlson, Gary Daniel, Peter Hunt, and Karl Lindberg. The authors also thank Harry Johnson and Ed Rotthoff for their comments.

REFERENCES

[1] Hunt, P., and Carlson, C., "A Data Flow Architecture for Multi-Mode Radar Simulation", Interservice/Industry Training Systems Conference, Orlando, FL, November, 1988.

The CCTT Development Approach: Integrating Concurrent Engineering and User-Centered Development

Tom Mastaglio, PhD, CCTT Training Effective Advocate
Don Thomson, CCTT Development Manager
IBM CCTT Integrated Development Team
Orlando, Florida

ABSTRACT

The Close Combat Tactical Trainer (CCTT) is being developed using a concurrent engineering approach that organizes the engineering effort into integrated teams assigned by major system components or products. These integrated teams include industry representatives, Army systems materiel development engineering staff from STRICOM, and TRADOC user representatives. Some users are assigned to an integrated development facility which is located near the material development customer so that they can interact on a daily basis with the engineering staff. Others are proponent level subject matter experts assigned to normal training and combat development jobs at the Army proponent schools and centers but readily accessible to the engineering staff. A key member of the development team staff, the CCTT Training Effectiveness Advocate, has as a primary responsibility implementing this concept.

CCTT development requires a strong user focus because it is a complex training system with a primary product of improving human performance. Major training system development efforts, like CCTT, must focus on human performance as a product rather than as merely one consideration in determining overall system effectiveness. This requires the development effort to have a user-centered design focus. Furthermore, CCTT is an extremely complex Human-Computer Interaction system. The combination of these two factors resulted in a joint industry/government decision to include field users up-front in the design and development phase of the program.

ABOUT THE AUTHORS

Dr. Tom Mastaglio is assigned to the IBM CCTT Integrated Development Team as the Training Effectiveness Advocate. He retired from 22 years of active duty with the U.S. Army in 1991. His last assignment was as a training developer at Headquarters TRADOC specializing in training simulations, aids, and devices. He holds a doctorate from the University of Colorado in Computer Science. His research interests include human-computer interaction and training effectiveness in simulation-based training systems.

Mr. Don Thomson is the Manager of CCTT Integrated Development in Orlando, Florida. He is responsible for all engineering and development activities associated with CCTT. Mr Thomson has been a manager of engineering organizations for IBM for 26 years. His most recent assignment was Manager of Strategic Development for IBM Federal Systems Company in Manassas, Virginia, where his focus was developing world-class technologies, organizations, and processes.

INTRODUCTION

The United States Army requires training environments in which to practice combined arms close combat in order to provide collective training for combat arms units and their supporting elements when organized into a combined arms team [1]. Based on the SIMNET [2,3] proof-of-principle demonstration that the synthetic environments provided by Distributed Interactive Simulation (DIS) can accommodate this requirement, the Army established a requirement for the Close Combat Tactical Trainer (CCTT).

CCTT is a Distributed Interactive Simulation [4] system wherein simulated elements replicate combat vehicles, weapons systems, and command and control elements networked for real-time, fully interactive collective task training on computer generated terrain. The CCTT system will primarily support maneuver company and battalion commanders in planning, conducting, and reviewing unit training on a free play, computer-generated synthetic battlefield. Contractor personnel provide site support and assist as directed by the training unit commander.

In order for CCTT to move beyond the success of SIMNET and insure all human performance and training requirements are accommodated during design, a user-focused development approach was required. The Integrated Development Team (IDT) uses concurrent engineering [5, 6] but further is incorporating prototypical system users into that process. The user representatives which participate in the concurrent engineering process are known collectively as a User Optimization Team, more specifically for CCTT, the Army Optimization Team. That team is comprised of both on-site users assigned from the Army's Training and Doctrine Command (TRADOC) and a supporting cast of Subject Matters Experts (SME) working in development assignment at the TRADOC schools and centers.

We will describe the integrated development process, how users are being integrated into that process, and our initial experiences with the CCTT development effort. Our purpose in offering this description is to share with the training systems development community what we believe is an innovative approach to addressing the requirement to incorporate the users perspective. This is a requirement which is becoming increasingly crucial as distributed interactive simulation systems get developed which are highly complex and include more realistic representations of combat systems, as well as a synthetic replication of the environments in which they must operate.

WHY A USER-FOCUS IS NEEDED

CCTT is an extremely complex human-computer interaction system because of the variety of types of interfaces it includes. Included in its system design are workstation interfaces to the computational system, simulator interfaces to the synthetic environment, plus computer interface devices and techniques that do not necessarily replicate a real world piece of equipment, but which must provide realistic access to the synthetic environment. We want to insure that the functional complexity of CCTT does not carryover to the manner in which the actual interfaces appear to their users. Therefore, a participatory approach that includes [7] a user-centered focus is essential.

The entire purpose of CCTT is to train soldiers to better perform their combat tasks as a team. It is not unique, but rather representative of an entire class of training systems which will be developed in the next several decades to support the trend in military training toward augmenting operational exercises with simulator/simulation-based training. This class of systems present a unique problem in that their immediate product is not combat efficiency, but rather training effectiveness. The latter is a determinant of the former of course, however the success or failure of these training simulation systems will be:

- their user acceptance,
- ability to replicate real world task performance conditions, and
- efficacy in changing human behavior (i.e., enhanced task performance proficiency in both replicated and actual environments)

User Acceptance

Development activities need to include the participation of actual users of these training systems to guide design activities toward meeting these three objectives. User acceptance will be enhanced by garnering the consultation of prototypical users who live and work in the training/combat development

communities throughout the design and development process.

There is, of course, a direct impact on the system design resulting from user participation. There is also an indirect impact, it is on user acceptance. User acceptance is approved when the user community learns that a system development effort is concerned about what that community thinks and wants.

Replicating Task Conditions

Replicating the environment in which training occurs is difficult, especially insuring that we include as part of a selective fidelity analysis [8] those aspects which are critical to performing the collective tasks. Users must be consulted when it comes to developing the virtual environment used in a training simulation. Similarly, although the fidelity of each simulator should map into a needs analysis based on those collective tasks, users have to be called in to review that analysis and subsequent design decisions.

Exposing design descriptions and prototypes of simulators and workstations to users to insure they include features that are crucial to performing the training tasks so that learning will be accommodated. Selective Fidelity is a key concept for designing DIS systems: the "selection" process has to be user-supported and training task focused. Both these require user support for design.

Improving Task Performance Proficiency

Effectively modifying human behavior is both the most crucial of these three criteria and the most difficult one to evaluate during design and development. The expert judgment of users who are current in their combat discipline needs to be brought to bear in evaluating design decisions.

Similarly, this type of user needs to be called on to assess system effectiveness incrementally during development and early production runs. This evaluation could be viewed as either a precursor to required operational testing or, if properly organized and configured, as an alternative.

A training system, more so than even men-in-the-loop combat systems, must be developed with a user focus. Training effectiveness will be achieved only if user needs and expectations are understood and incorporated early in the development process. Our approach to CCTT was established with that in mind. We are using an integrated development methodology which includes users representatives as part of an industry/government team.

THE USER OPTIMIZATION TEAM

We call our approach to integrating users into the CCTT development a User Optimization Team, specifically the Army Optimization Team because the Army is the customer. It is comprised of both an on-site team of soldier-experts and a matrix of outside expertise, other active duty soldiers available for off-site and on-site consultation. The industry team designated an engineering staff member whose primary role is overall responsibility for integrating users, the Training Effectiveness Advocate.

On-Site Team

The on-site team is comprised of three combat arms soldiers who represent the primary training audience for CCTT. The Team Leader is an Army Officer with Company Command experience. He is the functional expert on the overall use of the system to train collective tasks at the platoon and company level.

Two senior Non-Commissioned Officers (NCO), one from the Armor, the other the Infantry Center provide expertise at the simulator and task performance level. The NCO's are master gunners and experienced training developers. They are members of Concurrent Engineering (CE) Teams, attending weekly meetings, reviewing design documentation, and advising system and software engineers on decisions regarding the system design. The on-site team is also responsible for interfaces between the CE Teams and outside subject matter experts.

Subject Matter Experts

A network of Subject Matter Experts (SME's) was established by the Army's requirements integrator for CCTT, the TRADOC System Manager for Combined Arms Tactical Trainers (TSM-CATT). Twenty-five SME's at 11 proponency centers were designated. These SME's are training and combat developers whose normal job is teaching, developing, or evaluating tactics, techniques and procedures for

their center. Their areas of expertise include combat operations within their branch, the use of specific pieces of equipment on the battlefield, combat service support operations, etc...

SME's serve as a resource to answer specific questions and are called on to support on-site user reviews. They are all available via an electronic mail connection. Queries regarding system design issues are forwarded through the on-site members of the optimization team to them via that system. When user review panels are required to assess analytic efforts, preliminary designs, and prototypes, these same personnel are assigned temporary duty at the development facility.

USER CENTERED DEVELOPMENT AND CONCURRENT ENGINEERING

The CE Teams developing CCTT are product focused. STRICOM's engineering staff and industry development engineers comprise these teams. The teams include expertise from a variety of domains that may or may not also be represented within the program by joint working groups. Users, who are really the customers of STRICOM, are also members of the CE Teams. Additionally, they serve as needed on subteams developing individual components or focusing on special programmatic or technical issues.

Concurrent Engineering Teams

The CCTT Integrated Development approach organizes all of the engineering effort and personnel into four CE Teams shown in Figure 1. A Module Team is responsible for the design, prototyping and production of all simulator modules in CCTT. A Semi-Automated Forces (SAF) Team is responsible for designing and implementing the SAF software. A Workstation Team is responsible for software development and hardware integration of all workstations in CCTT. This includes both those used by the training audience (e.g., the workstation replicating the Fire Support Element) and those used to support system operations (e.g., the Master Control Console). The Architecture CE Team concentrates on architecture issues and products (e.g., network, databases, models, PIDS, etc.).

The System Integration Team is not identified as a CE Team per se but consists of the leads from the CE Teams and Working Groups. It integrates system level issues for the other four teams.

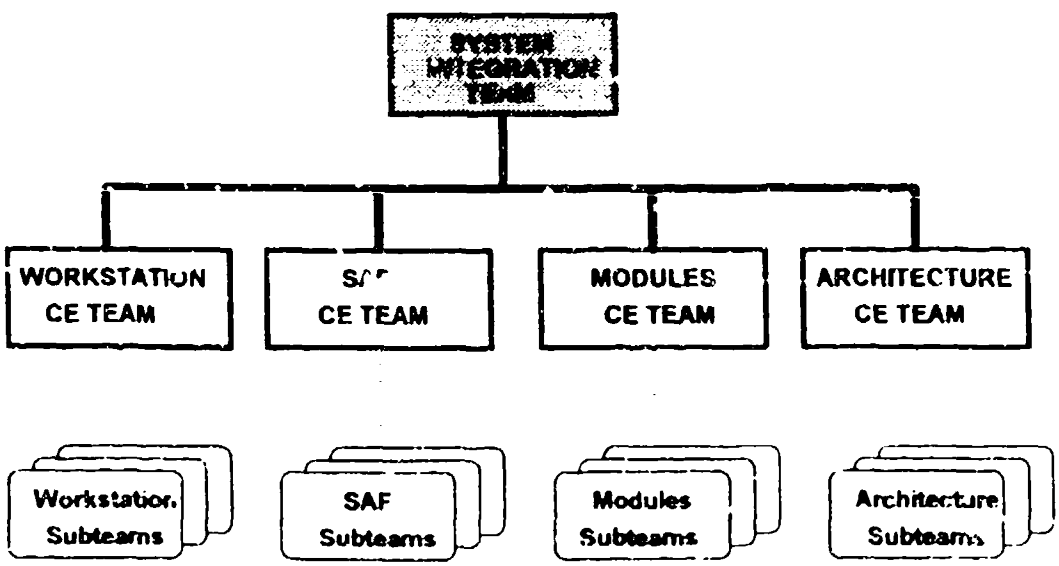


FIGURE 1 CE Team Organization for CCTT Integrated Development

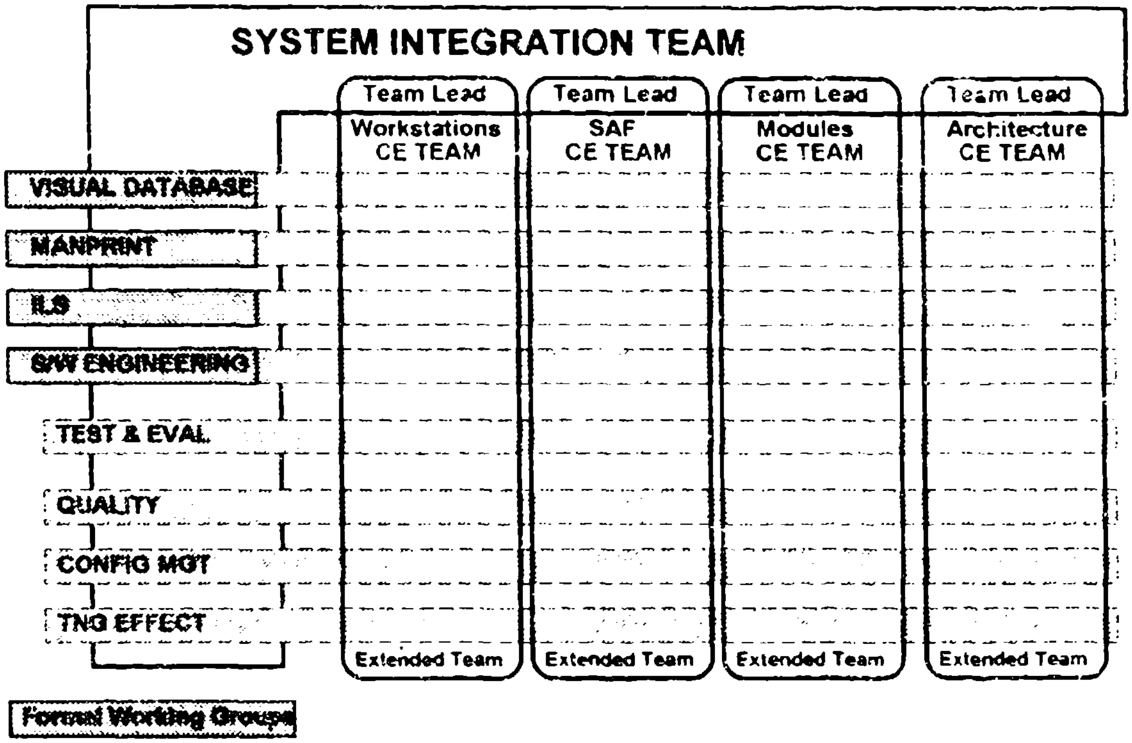


FIGURE 2 Relationship Between Functional Domains and CE Teams

Sub teams are comprised of members of each CE Team. They work on a specific piece of hardware (e.g., the M1A1 tank) or software (e.g., BLUEFOR tactical expertise combat instruction set). These teams will meet as required and are often less formal in that they are frequently those engineers who work together, perhaps even in the same office, on a daily basis on some part of the system. The four formal CE Teams meet weekly on a scheduled basis.

As shown in Figure 2, the CE Teams have functional expertise support across a variety of domains. Some areas of expertise reside in one or two individuals (e.g., Safety) and therefore these same individuals participate on all CE Teams. Other areas require a large supporting staff which is split between teams (e.g., Software Engineering). Teams are collectively responsible for addressing the requirements and concerns of each domain, they call on the assigned team member in that area to support them. Each team member is in part responsible for the team's solution. This is a departure from many development organizations where staff from some areas serve to oversee and critique the efforts of others rather than directly contributing to the solution.

Role of User Optimization Team in CE Process

The User Optimization Team is a crucial part of each team. Its members provide an interface to and represent the proponent and user community for CCTT. Their role encompasses serving as a respondent to requests and inquiries that arise during CE team activities and serving as a catalyst for user-focused activities.

As an example of their role as a respondent, a question about the potential application of CCTT in training or how a piece of equipment being simulated (e.g., the Fire Support Team's Laser Designator) is actually used, gets referred to the User Optimization Team member present. He may answer the query directly if it falls within his area of expertise. If not, he will initiate an action to obtain an official answer from the respective proponent. He is responsible to the CE team for obtaining an adequate response. One must keep in mind that all such information is advisory only; the responsibility for what is implemented in CCTT still resides with the development engineers. Acceptability of the design and the

system during reviews and testing are still the responsibility of the industry engineering staff.

The Army Optimization Team members often find themselves serving as catalysts in that they need to help the CE teams identify when user reviews are appropriate. For example, coordinating user review of analytic results or prototypes scheduled for completion needs to be projected far enough in advance that the Army can provide suitable support. The Optimization Team finds that their role is often more proactive, reminding their CE Teams to plan for and request needed support.

EXPERIENCES AND OBSERVATIONS

As of this paper preparation, CCTT is completing its first six months of development and we have primarily initial experiential results to report. Customer support for integrating users into a development process is crucial. Selling the concept to the user community is also no trivial task. Lastly, helping engineers who are more comfortable with traditional methods understand why users are involved and the potential benefits of that involvement is challenging.

Customer Support

The material developer for CCTT is STRICOM. They had to agree with and actively support the User Optimization Team approach. Since for CCTT, the CE process and integrated development approach were also new, adding user participation has not been a major problem from an acceptance perspective.

There were initial growing pains, in that some government engineers and program managers were apprehensive with development engineers being able to communicate too openly with the user community. To insure that these opportunities were not abused we initially established "rules of engagement" for dealing with the Army Optimization Team. As the program progresses and the three communities (industry, government, and user) became more familiar with one another, confidence and trust has increased to the point where this is no longer a major concern

User Community Support

Staffing the User Optimization Team has proven to be challenging. Of the three on-site personnel, one was assigned for duty 3 months after contract award, the second 7 months after. The arrival of the third is pending.

A major challenge has been communicating the concept through the Material Developer and the System Integration Office to TRADOC schools. Another hurdle is the administrative burden imposed by recent personnel cutbacks which limit the number of soldiers available for assignment to fill this type of position. Several months lead time is needed to relocate a soldier on a permanent change of station.

To overcome these problems we recommend that government procurement agencies decide early, perhaps even during RFD preparation, certainly prior to contract award, that they want to integrate the User Optimization Team concept into a development effort. They should initiate the administrative actions required to identify and assign appropriate user community personnel to the industry team shortly after contract award.

Industry Support

Training industry engineers on how to leverage the User Optimization Team concept has been a challenge, but one where direct interaction with the individuals makes it easier to communicate (as opposed to having to overcome time consuming administrative and bureaucratic hurdles).

We initiated a parallel effort shortly after contract award to familiarize our industry engineering staff with Army operational and training concepts. First, we offered this information in briefing format. This was followed up by visits to TRADOC installations to learn about equipment and tactics from soldiers directly. This effort made the engineers aware of the valuable role an in-house expert could serve and helped them overcome any inhibitions about dealing directly with soldiers. We made the engineers feel that they are part of the Army team.

CONCLUSION

Based on our initial experiences, we believe a User Optimization Team is a recommended technique for incorporating user desires and perspective into a training system development program. We believe it is better than techniques such as user juries by themselves, or traditional periodic formal reviews for the user community and structured working groups.

The Army Optimization Team approach is a key aspect of our Total Quality Management (TQM) approach. It insures that user/customers for CCTT are an integral part of the development process with a goal of improving the quality of the product and ennancing user acceptance. Although we are using the approach in a training system and have argued above that the very nature of training devices make their development a pnme candidate for this approach, we believe that military (and for that matter most commercial) complex system development efforts should use a similar approach.

REFERENCES

1. IBM, The Close Combat Tactical Trainer Operational Needs Document, June 1, 1993.

2. Thorpe, J.A. "The New Technology of Large Scale Simulator Networking. Implications for Mastering the Art of Warfighting." Proceedings of 9th Interservice/Industry Training Systems Conference. pp. 492-501. American Defense Preparedness Association, Washington, D.C., 1987.

3. Alluisi, E. A. "The Development of Technology for Collective Training: SIMNET, a Case History," Human Factors, 33(3), pp. 343-382, 1991.

4. Beaver, R., O'Brien S., and Riecken, M., "Advanced Distributed Simulation Technology System Definition Document", LORAL, ADST Program Office, Orlando, FL, ADST/WDL/TR--92-03017-QTR2YYR2, 22 October 1992.

5. IBM Corporation, "Owego Guidance Document for Concurrent Engineering", Document Number 92-NA5-001, Federal Sectors Division, Owego, NY, February 24, 1992.

6 IEEE, Special Issue on Concurrent Engineer, IEEE Spectrum, July 1991.

7. Schuler, D. and Namioka, A, "Participatory Design: Principles and Practices", Erlbaum, Hillsdale, NJ., 1993.

8. Miller, D. C. "The SIMNET Architecture for Distributed Interactive Simulation," Paper presented at Summer Computer Simulation Conference, Baltimore, MD, 1991.

RECOMMENDATIONS FOR FRANCHISING AIRCREWS IN SYNTHETIC ENVIRONMENTS

Lynn A. Carroll
Air Force Armstrong Laboratory
Mesa, Arizona

Dee H. Andrews
Air Force Armstrong Laboratory
Mesa, Arizona

ABSTRACT

Synthetic environments will become increasingly important to the military in the future. The capability to optimally blend virtual, constructive and real environments will become crucial not just for aircrew training, but for other military uses (e.g, test and evaluation, research and development, prototyping, tactics validation). Technical advances in networking will theoretically allow any site in the world to be linked into world-wide synthetic environments. Individuals and components from the Joint Chiefs of Staff down to the individual warrior will be able to access these environments. Senior leaders will interact through synthetic environments in much the same way they currently interact with theater and battlefield level assets during war. Therefore, this paper does not focus on issues related to franchising upper echelon users of synthetic environments. This paper expresses recommendations and considerations about what will be required to franchise aircrews at the lower end of the hierarchy.

In the zeal to create and use synthetic environments, operating concepts, access tools and aircrew training requirements may be over-looked. Aircrews already voice the concern that they are merely "training aids" for senior leaders in large-scale exercises. The problem stems from aircrews not being allowed to function as they would in combat. This paper describes concerns, recommendations and access tools that should be considered to make sure that aircrews are properly franchised in the use of synthetic environments.

ABOUT THE AUTHORS

Colonel Lynn A. Carroll is a Command Pilot with extensive experience in tactical fighters and training aircraft (AT-33s, A-7Ds, T-39s, AT-38s, and F-15s). He is currently the Division Chief of the Aircrew Training Research Division, Armstrong Laboratory. Col. Carroll graduated from the University of Iowa with a BA and MS and was commissioned through the AFROTC Program. He served as the Commander of the 604th Direct Air Support Squadron at Camp Red Cloud, ROK, from 1987-1988. He also has served at all staff levels including 9th AF, HQ TAC, and the Air Staff. At the Pentagon, Colonel Carroll functioned as the Deputy Chief, Training and Warrior Management Division and as Chief of the Training Branch from 1988-1992.

Dr. Dee Howard Andrews is currently the Technical Director of the Aircrew Training Research Division, Air Force Armstrong Laboratory. He previously was a research psychologist for the Army Research Institute and the Naval Training System Center, both in Orlando, Florida. He received his Ph.D. in Instructional Systems from Florida State University. His research interests include; instructional systems development, training device design and use, training system evaluation, and training cost-benefit analysis.

RECOMMENDATIONS FOR FRANCHISING AIRCREWS IN SYNTHETIC ENVIRONMENTS

Lynn A. Carroll
Air Force Armstrong Laboratory
Mesa, Arizona

Dee H. Andrews
Air Force Armstrong Laboratory
Mesa, Arizona

INTRODUCTION

The Department of Defense has embarked upon a long-term program to make modeling and simulation (M&S) integral to future defense programs. It will be used in a wide range of roles from requirements definition, weapons systems development and acquisition, test and evaluation to training and education. The term for this initiative is "synthetic environments" (SE). This use of technology is deemed so important that the Department of Defense Office of the Director of Defense Research and Engineering has made SE one of only seven R&D "Thrusts" that it is pursuing.

Technology, real-world constraints and new age thinking about modeling and simulation have combined to significantly influence the creation and use of synthetic environments. SE will be used to provide and expand training opportunities that will complement and supplement future wargaming, exercises and operations tempo. When combined with live training, constructive (wargames and exercises) and virtual training simulations will provide a synergistic representation of what has heretofore only been available in battle.

Technical advances in networking will theoretically allow any site in the world to be linked into world-wide synthetic environments. Individuals and components from the Joint Chiefs of Staff down to the individual warrior will be able to utilize these new environments for their own purposes. Senior leaders will be able to interact through netted confederations of synthetic C4I environments in much the same way that they currently interact with theater and battlefield level assets during combat. Higher aggregate levels of constructive simulations with semi-automated forces and intelligent adversaries will meld with virtual and real weapons systems. This will provide decision-makers with the opportunities to apply and hone warfighting skills at the strategic, operational and tactical levels of war as well as supporting the decision-making process.

Invariably, man-in-the-loop simulation will be desired, if not required, to fully exploit realistic environments for decision-makers and warriors alike. As individuals and teams are transported into higher aggregate levels of simulation, concerns will arise regarding the relevance to those individuals' needs.

In the zeal to create and use SE for everything from requirements definition to manufacturing, it must be tempered by a requirement to do so. This creates a dilemma for leadership. Attempts to develop inclusive simulations that cut across various levels and types of simulations have been viewed as having limited utility, especially for the individual. Classic concerns of using individual warriors as "training aids" for those above them have created an aversion to participation in simulations and exercises alike. Serious problems can arise if lower-level SE users believe they are not allowed to function as they would in battle.

Incorporation of all levels of participants is not necessarily required or desired in every simulation. However, it may become extremely beneficial and cost-effective if a synthetic environment can support multiple opportunities to train while higher echelons practice the business of war at their levels. It

is imperative that the requirement to support higher levels of simulation with aircrews-in-the-loop accommodates the individual warrior's needs. It will be critical that the requirements of those above are transparent to those below.

Synthetic environments will offer the potential for increased training effectiveness, reduced training costs, accelerated mission readiness and ultimately improved combat effectiveness. Desert Storm showed that while the Services have dramatically improved their ability to fight individually and collectively as a joint/combined force, training opportunities still lag total combat requirements. Visions include linking ranges and thousands of weapon system platforms (both real and virtual) to provide more realistic training. While real platforms will continue to provide the operations tempo required to mature the force, virtual tools will provide those opportunities or alternatives unavailable due to cost, environmental, security and safety constraints and encroachment of airspace and ranges. This combination of training capabilities will provide a training synergy that is currently not available or readily accessible to the aircrew.

However, the utility of these simulations will rest with the ability to franchise the aircrew both when the aircrew needs are being supported by SE, and, as importantly, when the aircrew is used to support higher level requirements. Concepts and access tools must be developed that will meet the needs of the aircrew while supporting a broader use of synthetic environments.

TECHNOLOGIES THAT MAKE SYNTHETIC ENVIRONMENTS POSSIBLE.

Enabling technologies are having the dual effects of making simulation more acceptable and credible, while at the same time significantly reducing the costs of simulation. Classic training constraints and encroachment of live training environments have accelerated the need to develop training alternatives. New weapons systems and Global Reach-Global Power considerations are expanding training requirements. These enabling technologies are supporting an explosion in the application and utility of synthetic environments for all warfighters. Technology rollover is occurring more and more frequently as costs continue to drop significantly as computer speed and power increase dramatically. This relentless evolution has created opportunities for technology to reduce procurement and sustainment costs while dramatically increasing capability, realism and accessibility of training. The creation of realistic simulated environments, whether benign or full combat scenarios, will provide aircrews with new and expanding training opportunities. These realistic, simulated combat environments are becoming available and affordable.

Full visual systems supported by common, universal data bases can be networked locally or long-haul, secure, to bring all supporting elements together. Standardized, secure network protocols, high speed-high capacity nets and universal network interface units will allow aircrews to participate in any level or combination of simulation from live to virtual to constructive from their unit training device or while in their aircraft. Advanced image generation and display systems will make simulations realistic enough for full mission rehearsal or total combat immersion.

THE NEED TO FRANCHISE AIRCREWS IN SE

Despite their advantages, synthetic environments and simulation in general, do not enjoy universal support from aircrews and the current leadership. The reasons are varied, but focus on a lack of credibility, fidelity, cost and concurrency of simulation, especially for the fighter community. Also, concerns remain that simulation will again adversely impact operations tempo as in the 1970's when flying hours were reduced based on the potential for simulation to offset the cost reduction effort. Consequently, it is imperative that the aircrew be fully franchised

in new and expanding simulation applications.

The utility of synthetic environments rests with the ability to "franchise" the aircrew both when the aircrews' requirements are being specifically supported by SEs and, as importantly, when the aircrew is used to support higher level objectives.

Webster's New Collegiate Dictionary defines franchise as "freedom or immunity from some burden or restriction vested in a person or group." In the case of synthetic environments, the burden or restriction would be the concern that aircrews would be constrained in their ability to fully interact with a simulated battlefield compared to their wartime mission requirements To be franchised, the aircrew must be "empowered to obtain value from their involvement in a training environment." This empowerment will come from establishing legitimate simulations and providing acceptable access tools based on aircrew training requirements.

Because of the nature of their positions in the chain of command, and because of the relative ease in representing the type of equipment, communications and data they would use in a theater-level battle, it should not be difficult for senior leaders to feel franchised in an SE.

However, a problem will occur if aircrews do not feel franchised (This problem may exist for all warriors at the lower end of the chain of command. However, this paper only addresses aircrew concerns). One primary concern is that the aircrews will serve only as "training aids" for leaders/decision-makers higher up in the chain. This concern is derived from experiences aircrews have had in large, live-fly, joint exercises in which participation was not conducive to good training. Loitering, flying designated ground tracks or acting as targets for ground-based systems in the simulated theater were not, and will not be seen as a desirable activity. Constraints that are necessary in the real arena need not be tolerated in simulation. Employing a "limited" weapon system may provide unrealistic and/or negative training. Aircrews are looking for an unconstrained, all switches up combat environment. Anything less will be viewed as an accommodation to someone else's requirements. A self-centered, but legitimate concern.

The potential consequences of disenfranchised aircrews will be manifested in several ways.

- Over-tasking of aircrews. The proliferation of SE may inundate the down-sized force structure. Single-seat aircraft in eighteen unit equipped squadrons may lack the manning to cover new additional taskings to participate in SE even if it is beneficial.

- Lack of relevance. Boredom or lack of attention may result if the simulation is not realistic enough or the aircrew perceives a training aid status. Scenarios must be stimulating and challenging and provide the aircrew training opportunities that are unavailable or readily accessible in his current regimen.

- Poor desired outcomes. If aircrews are not properly franchised, training benefits will not be achieved by the aircrews or by the higher echelon participants. If the aircrew is supporting a higher level simulation, those objectives may be skewed due to the poor performance by the aircrew and represented weapon system. Poor performance may produce bad data which may negatively impact decision-making by higher echelon leaders.

- Dislike of SE. Disenfranchised aircrews could feel some resentment that they are "forced" to spend time in SE. This may be true even if their access to an SE is via their actual aircraft. This resentment could carry over from training applications of SE to other applications such as: test and evaluation, weapons systems design, and tactics development where M&S credibility is already a problem.

- General negative attitude towards simulators. Disenfranchisement will be most

probable for aircrews accessing an SE with a marginal access tool (simulator). They will be frustrated by being asked to accommodate "sim-isms" and limited fidelity that constrain their ability to employ their weapon system as they would in combat. Classic negative training. Less apparent will be the reticence of aircrews who view current simulators as somehow threatening. The windowless building with meatlocker temperatures. Cipher locks controlling access to nonconcurrent devices with, at best, a window to the battlefield. Canned scenarios from someone else's database. Grade sheets made out by contracted instructors. The dreaded cry from the ops counter, "We need somebody for the sim" as if it were time to feed the beast as opposed to an opportunity to hone combat skills.

- Paranoia concerning flying hour tradeoffs. Until aircrews are effectively franchised in SE or any simulation, inordinate concern will exist regarding potential loss of flying hours to simulation. The concern is real, it happened in the 1970s. The solution is to provide training opportunities that do not overlap or compete with flying time by pursuing low-cost, high fidelity access tools and realistic synthetic combat environments.

RECOMMENDATIONS FOR INCREASING THE LIKELIHOOD THAT AIRCREWS WILL BE FRANCHISED IN SE

In order to provide adequate synthetic combat environments and access tools that have the proper functional and physical fidelity, we believe it will be necessary to implement the following recommendations that are aimed at franchising aircrews. We believe that the consideration of the following recommendations would be very beneficial.

1. Provide realistic, accessible simulated combat environments that provide training opportunities that are currently unavailable to the aircrew. (e.g, full mission training, force-on-force engagements, kill removal, electronic warfare)

2. Provide multi-use, affordable, high fidelity training devices in the unit that can stand alone, net with others or access SE.

3. Use aircrews in SE only when they are productively engaged in a specific mission. We are concerned that aircrews may become disenfranchised if they are participating in an SE where they are only used to support the objectives of the exercise.

4. Design SE and SE access tools based on the needs of the aircrew in order to fully franchise the aircrew and ensure support of the SE objectives.

5. Franchise aircrews in the "Big Picture" through briefing and debriefing sessions at all levels. SE should provide the technology to make briefings/debriefings easily available for all levels. Long haul, secure networking of simulators, constructive wargames, and range instrumentation debriefing centers will ultimately allow warriors at all levels to be linked. Aircrew distance from the primary SE exercise areas before and after the exercise will not be a problem for briefings/debriefs. Quality video teleconferencing will go far toward allowing the aircrews to feel that have been franchised in the entire process.

6. Ensure senior-level users address the concerns of the aircrew.

PROVIDING ADEQUATE ACCESS TOOLS FOR THE AIRCREW

The capability will soon exist to effectively and affordably immerse aircrews <u>in</u> the synthetic combat environment as opposed to merely a view <u>of</u> the battlefield. This immersion will require that legitimate access tools be made available to transport the aircrew into the simulation to fully leverage this expanding training capability. Training devices must evolve to become the access tools to these environments. As such, they will be required to fully replicate the capabilities of the weapon system they represent while porting the aircrew into the simulated combat environment. The

simulation also must provide for mission requirements such as multiship employment while including combat support assets such as AWACS and EW platforms to ensure realism.

Three types of fidelity must be addressed to ensure appropriate support to the aircrew. Functional fidelity is based on the faithful representation and operation of the simulated system and subsystems at the model and integration level. Physical fidelity requires that the interface between man and machine provides adequate familiarity to preclude "sim-isms". Both of these relate to concurrency with the weapon system. Psychological or perceptual fidelity, while less precise, requires that the aircrew "feels" right in the simulation. Shortcomings in any of these could result in negative training, lack of confidence or credibility in the simulation or poor results that could skew decision-making. Reduced or selective fidelity also may require a validation of training effectiveness for which adequate studies are time-consuming and difficult at best.

Rather than pursue the question of "How much fidelity is enough?" We must pursue technology, methods and applications that will make full fidelity affordable and available. While full fidelity may not be required for all applications, operating in a full combat scenario makes it imperative.

However, some have suggested that the aircrew need only be provided with a minimalist solution primarily to reduce costs. The "60% solution" may allow the aircrew to believe they are in the synthetic environment because it "generally looks right", and it properly represents the data base. However, if it doesn't allow the aircrews to adequately perform their combat duties and tactics, their frustration may be similar to that caused by lack of concurrency in their current genre of training devices.

While the SE community is investing large amounts of resources in developing and demonstrating the M&S and networking infrastructure necessary to create SE, aircrews are offered "60% solutions" for their simulated weapon system. This is akin to building an Indianapolis speedway, then providing the race car drivers with go-carts to simulate the Indy 500. Aircrews want to conduct effective training that cannot be provided elsewhere, whether it be in the aircraft or the ground-based training system. They want to enter combat, real or simulated, with a full-up system, and not be constrained by lack of fidelity.

In order to "train the way we intend to fight", the synthetic training environment must represent the combat environment without peacetime constraints and with minimal concessions to reduced fidelity. The simulated weapon system must be able to conduct the specific mission without accommodations. Therefore, the access tool must fully support the warfighters requirement to enter the simulated combat environment with his total weapon system and that of his wingman or total force package. If the requirement for aircrew-in-the-loop is to operate in a simulated combat environment, then the aircrew must be provided with a full-up simulation of his weapon system. The corollary is that if higher levels of simulation require a weapon system be involved, then aircrew-in-the-loop is required since the aircrew is an integral part of the weapon system. Anything less will not fully franchise the aircrew and will require arbitrary extrapolation to determine value and utility to training and decision-making objectives alike. Aircrew requirements must be used to determine the capabilities of the training device or access tool, not the objectives of the supporting SE. Therefore, in order to fully franchise the aircrew, the requirements of the aircrew must be met before the objectives of the synthetic environment can be achieved.

CONCLUSION

Franchising the aircrew, if not properly understood and addressed, could greatly reduce the effectiveness of synthetic environments. The SE development community may be able to make huge strides in the technological challenges related to SE

(e.g., networking, data base development, instrumentation). However, if aircrews and other warriors at the bottom of the chain are not accommodated, they may discount the training value of the SEs in which they participate.

SE proponents must avoid the following attitude about franchising the aircrew: "The folks at the bottom will just have to get used to the idea that in a theater-level SE their needs and concerns are not of paramount importance. They will get something out of their participation regardless of how they feel about it. They are paid professionals and they will just have to do what they're told."

Such an attitude will not help SE attain their full measure of utility. In this paper we have made recommendations that we believe will allow the full enfranchisement and empowerment of the aircrews that will be such important parts of future synthetic environments.

An aircrew should not be expected to enter combat, real or simulated, with less than a full-up weapon system and the full complement of combat suppo[illegible]. If synthetic environments have anything to offer airc[illegible]ws, it is an unconstrained opportunity to "train [illegible]e way they will fight." If they are constrained, then the simulation has failed and we have failed to support the aircrew.

WHAT IS ISO 9000, CAN IT PLAY A FUTURE ROLE IN TRAINING ACQUISITION STRATEGIES?

James B. Coughlin
Reflectone, Inc.
Tampa, Florida

ABSTRACT

Due to the wide spectrum of products and services acquired by the training community, it is virtually impossible to develop a common denominator for contractually assuring the adequacy of the products and services purchased. This situation is additionally complicated by the fact that training acquisitions often involve a significant level of non developmental items (NDI) equipment or the purchase of services.

Debates of the recent past have questioned the adequacy of MIL-Q-9858 as the most effective procurement tool in assuring this wide range of training community needs. Recommended alternatives such as "best commercial practice" have equally raised concern among procurement leaders as being too vague and not universally defined.

Could the ISO 9000 series of specifications help solve this problem? Could it become another tool in developing an effective acquisition assurance strategy? This paper will define the ISO 9000 series of standards, and provide an analysis of how these standards could effectively be used (once approved for use) by the training community as an acquisition assurance tool.

ABOUT THE AUTHOR

Jim is currently in the employ of Reflectone, Inc. as a Senior Quality Engineer. During the nine years Jim has served with Reflectone, he has been assigned to a wide range of Product Assurance responsibilities. Specific responsibilities have included Manager of Software Quality Engineering/Software Configuration Management for the C-5 Air Refueling Part Task Trainer, the AV-8B suite of Maintenance Trainers, the E-6A Operational Flight Trainer (OFT), the EA-6B OFT, and the CV-HELO OFT. Subsequent assignments have included Program Quality Engineer for the EA-6B Controls/Flight/Tactics upgrade, logistics provisioning, the BAe 125-800/1000 OFT, the C-130H OFT, and team leader of Reflectone's ISO 9001 TQM preparation team. Prior to Reflectone, Jim was employed with E-Systems, Sperry Gyroscope, and the U.S. Army's STRATCOM command.

Jim has obtained a Masters of Aeronautical Science Degree from Embry Riddle Aeronautical University, Daytona Beach Florida and a Bachelors of Business Administration - Finance from the University of South Florida, Tampa, Florida. Jim is also a proud graduate of the U.S. Army's Signal Corps School, Fort Monmouth, New Jersey. Additionally, Jim is a graduate of an IQA registered ISO 9001 Lead Quality Assessor course.

Jim has been a member of the American Society of Quality control for the past eight years and is currently pursuing his Certified Quality Engineer designation Jim has submitted his ISO Quality Assessor (provisional) application for consideration by the Registration Accreditation Board and the Institute of Quality Assurance.

WHAT IS ISO 9000, CAN IT PLAY A FUTURE ROLE IN TRAINING ACQUISITION STRATEGIES?

James B. Coughlin
Reflectone, Inc.
Tampa, Florida

Why the Acquisition Needs of the Training Community Are Unique From a Quality Assurance Stand Point?

The training community's needs are diverse in scope and the technologies that they span. A turnkey training system includes supporting elements such as simulator hardware, software, courseware, aircrew and maintenance training, and simulator maintenance services. Although the training community has historically established an excellent reputation for the delivery of proficiently trained personnel, the continued emergence of new technologies and dwindling procurement resources creates future challenges for all levels of training management. As the training community continues looking for methods of self improvement, there has emerged a new quality assurance tool worth considering. This tool may represent a basis for integrating under a single assurance standard all elements of a training system. The author suggests such integration would improve management's ability to cost effectively deliver high quality products and services. Clearly, such a tool would be beneficial to the training community.

Surveying the existing quality assurance standards which are employed by the training community, it becomes readily apparent that the type of tool employed is dependent upon the specific element of the training system which is being purchased. We find that each element uses a different quality assurance standard. Table 1 provides an illustration of the specific quality assurance tools used by the training community based on the training system element purchased.

Table 1 Existing Use Of Quality Assurance Standards In Training Systems Acquisitions

Acquisition Element	Standard Applied
Simulator Hardware and Software Design	1) MIL-Q-9858A, (DOD-STD-2168) 2) "Best Commercial Practice"
Training Program Development	1) MIL-STD-1379D Paragraph 4.4
Training Instruction	1) MIL-STD-1379D Paragraph 4.4 2) Command/Local Practice
Maintenance Services	1) MIL-I-45208A 2) "Best Commercial Practice" 3) Command/Local Practice

The posture of MIL-Q-9858A limits its full scale application in training community acquisitions:

Thinking of Department of Defense acquisitions and quality assurance, we most often think of MIL-Q-9858A (Quality Program Requirements). When speaking of commercial acquisitions, we most often use the term best commercial practice (BCP). During its thirty years of life, MIL-Q-9858 has

played a traditional role of quality assurance for the factory applicable to hardware purchases and the production environment. There has always been some difficulty, and in recent years reluctance, of applying MIL-Q to non-hardware products and services. Additionally, many newer technologies used by the training community have emerged as non developmental items (NDI), for which little or no quality assurance tools are employed by the purchaser.

For those purchases not involving the factory, such as training programs and maintenance services, either alternatives to MIL-Q are used, such as MIL-STD-1379D (Military Training Programs) paragraph 4.4 for training programs, or often contractor/customer designated quality assurance plans are employed. Although each alternative has proven successful in the past, both of these alternatives tend to create a polarization of quality assurance practices into element unique systems, with little or no contribution for tying these unique elements into a centralized quality assurance strategy. I suggest that such polarization minimizes the impact of quality assurance as an effective management tool due to its failure to provide an integrated visibility into the training systems acquisition process.

Is the use of "best commercial practice" a safe acquisition alternative?

As acquisition streamlining has been discussed in the last several years, one quality assurance approach that has emerged is the use of "best commercial practice" (BCP). One of the major assumptions with this approach is to allow the contractor to define BCP and hold them responsible for the final outcome of the deliverables. Although this represents an attractive alternative in streamlining efforts, its application represents a concern for quality assurance professionals. What exactly is BCP? Is there any general consensus on its definition and meaning, is it easily put into practice?

The most extensive analysis of this question that I have come across was prepared by T. Tierney of the Link Flight Simulation Division of the Singer Company. As Mr. Tierney noted, "it has been Link's experience that quality obtained under best commercial practice ranges widely, from full compliance with MIL-Q-9858 to virtually no compliance whatever. Establishment of precise standards for best commercial practice leads to three desirable results: quality costs are reduced, suppliers know precisely what is expected, and minimum standards are defined for training systems in general. The absence of a precise definition makes the tasks of selecting, monitoring, and evaluating suppliers difficult and subjective".[1]

Link successfully analyzed a series of existing commercial and military based standards and generated the beginning of a consensus at both Link and their suppliers as to the definition of BCP. As with any new initiative, coordination was required among Link activities and locations as well as with suppliers and the procuring activities. The effectiveness of their initiative can best be measured in the content of a single company attempting to design and construct standards to suit the complexity and criticality of the system to be procured. This effort is to be applauded since it was initiated by the contractor to put in place an unspecified need of the customer, namely the quality assurance of major subcontractors and vendors.

Today, we have a standard available which clearly meets the needs that T. Tierney noted as necessary, namely a tailorable set of minimum requirements for quality assurance. Additionally, the lead time required for developing a consensus for this new standard is not required, as consensus is already substantial and global in dimension. And finally, all quality assurance practices for each training system sector can be integrated into this one common strategy. This will greatly enhance the ability of management at all levels of the procurement chain to more effectively manage using quality assurance. This standard is commonly know as ISO 9000.

What is ISO 9000?

ISO 9000 is a family of standards and guidelines which define the minimum requirements for establishing and maintaining a quality management

system. This family is generic in scope and language which creates a tremendous degree of flexibility in their application and use. A further investigation of the organization which issues the standards as well as the standards themselves is paramount to understanding the significance that the ISO 9000 standards may represent for the training community.

Who is the International Organization for Standardization (ISO)?

The issuer of all ISO standards is the International Organization for Standardization located in Geneva, Switzerland. Although ISO (pronounced eye-sch) is most often understood to stand for International Standards Organization, according to the current Secretary General of the organization, "ISO is actually a Greek word which means equal".[2] This word was selected by the organization because it represents the goal of all ISO standards, the "equalization" of standards for international use.

Therefore, expect to see "ISO" used as the acronym for the organization, and its meaning to be used both ways noted above. I have found the most common use of ISO in the United States is International Standards Organization, which many will argue is incorrect. The International Standards Organization and the International Organization for Standardization are not two distinct agencies, they are truly one and the same!

I will use the term ISO to refer to the organization as they do in their published literature.

Brief Background and History of ISO:

The ISO was founded in 1946 in Switzerland. "It is the specialized international agency for standardization, at present comprising the national standards bodies of 92 countries. ISO's work is decentralized, being carried out by 179 technical committees and 620 subcommittees which are organized and supported by technical secretariats in 34 countries. The Central Secretariat in Geneva assists in coordinating ISO operations, administers voting and approval procedures, and publishes the International Standards.

The membership of ISO is made up of "member bodies". Member bodies are the individual national bodies representative of standardization in that particular country. Typically, only one such national body is accepted for membership to ISO. Member bodies are entitled to participate and exercise full voting rights on any technical committee of ISO, and are eligible to participate in the governing bodies of the organization. An International Standard is the result of an agreement between the member bodies of ISO.

More than 70% of the ISO member bodies are governmental institutions or organizations incorporated by public law. The remainder have close links with public administration in their own countries. Some 450 international organizations work in liaison with ISO technical committees, including nearly all of the United Nations specialized agencies.

At any given time, the estimated number of people supporting the member bodies in developing International Standards is 30,000 engineers, scientists and administrator. To date, more than 100,000 people have participated in standards development. Individuals are nominated by ISO members to participate in committee meetings and to represent the consolidated views and interests of industry, government, labor, and individual consumers in the standards development process.

The purpose of ISO is to promote the development of standards in the world with a view to facilitate international exchange of goods and services, and to developing co-operation in the sphere of intellectual, scientific, technological and economic activity. The results of ISO technical work are published as International Standards.

For those not familiar with the International Standards published to date, review of the ISO Catalogue, is very interesting. The catalogue lists all currently published ISO standards. The standards have been categorized into no fewer than 39 fields and 450

groups. The list contains diverse fields such as banking and financial services, aircraft and space vehicles, health care technology, information processing systems, image technology, products of consumer interest, and road vehicles.

The scope of ISO covers standardization in all fields except electrical and electronic engineering standards, which are the responsibility of the International Electrotechnical Commission (IEC)."[3]

In this competitive day and age, each industry needs to employ synergistic approaches and global thinking in their improvement strategies. ISO's operating practices reflect a truly synergistic approach to standards development and are global in dimension as illustrated above.

American Participation In The Preparation of ISO 9000:

The American National Standards Institute (ANSI) is the member body representing the United States within ISO. For issues pertaining to quality assurance, ANSI delegates the responsibility to the American Society of Quality Control (ASQC), the most prominent society in the United States promoting the use and dissemination of quality assurance practices. ASQC represented the United States on the ISO technical committee (TC 176) that prepared the ISO 9000 series of standards.

An International Standard may be used as such, or may be implemented through incorporation in national standards of different countries. The American equivalent of ISO 9000 is ANSI/ASQC Q-90.

You probably already have had an exposure to an ISO standard. Next time you purchase a roll of film for the family camera, look at the box closely. An ISO film speed standard is now listed on every box of film sold in the United States. The use of ISO standards is closer than you think!

Where Did The ISO 9000 Family of Standards Evolve From?

When one studies the ISO 9000 family of standards they will note that they are similar to MIL-STD-9858 and the NATO-United Kingdom's AQAP-1. In fact, the ISO 9000 family of standards were based heavily on the "successor" of the above noted standards, British Standard 5750 published in 1979.

Is there any relationship to MIL-Q-9858?

As noted in the previous paragraph, the ISO 9001 standard is very similar to MIL-Q-9858. Several differences are significant for management when comparing the two. The differences between the two stem from their historical usage, and their interpretation by the user communities. Both of these factors limit MIL-Q and create limitless possibilities for the application of ISO 9000 series standards.

Of historical significance is the precedence lying in front of MIL-Q, over thirty years of use and interpretation. This history is not only the strength of MIL-Q, but also a factor contributing to its limited training-community utility (limited to the factory environment). If you think you can understand MIL-Q by simply reading it, you are wrong. To adequately understand MIL-Q, you must also possess an understanding of a significant portion of its precedence, most of which is acquired by exposure to its implementation. The historic application of MIL-Q, with its unapparent precedence, as well as its language, has limited its universal understanding and use as a management quality assurance tool throughout the training community.

Although ISO 9001 has a precedence, it has only been in effect for six to seven years. Learning the use of ISO 9001 is much easier. The utility of the ISO family is that it is not limited to a single production or service sector of industry, but is designed to be applicable to all sectors of production and services. For example, ISO 9001 could be used in defining the quality assurance plan requirements of MIL-STD-1379. Another unique aspect of ISO 9001 is its intrinsic value as a management tool; not at first readily apparent, but significant when placed in motion.

Why The ISO 9000 Family of Standards Are Unique Within ISO.

The ISO 9000 series of standards are the only ISO standards which deal specifically with management systems. Although quality assurance and quality

systems are often thought of as technical tools, they are most closely aligned with the management sciences and the control function of management. Let's take a look at the specific elements of the ISO 9000 family of standards:

- ISO 9000, Quality Management and Quality Assurance Standards - Guidelines for Selection and Use

  This guideline is used by management in selecting the ISO quality standard most appropriate for use in specific contractual situations.

  It is currently intended that only ISO 9001, 9002, or 9003 be invoked contractually. These documents are approved standards and not merely draft international standards (DIS), guidelines, or references as are the remainder of the other family members (to be discussed later).

- ISO 9001, Quality Systems - Model for Quality Assurance in Design/Development, Production, Installation and Servicing

  This standard is intended to be invoked contractually when the nature of the procurement involves not only production, installation and servicing work efforts, but also design/development efforts.

  Design/development work efforts are unique by their very nature and ISO 9001 provides for management controls in this predominately engineering arena.

- ISO 9002, Quality Systems - Model for Quality Assurance in Production and Installation

  This standard is intended to be invoked contractually when the nature of the procurement involves only production and installation related activities.

  A "build to print" effort would best be managed with this standard. Applicability to NDI suppliers would also be appropriate.

- ISO 9003, Quality System - Model for Quality Assurance in Final Inspection and Test

  This standard is intended to be invoked contractually when the scope of the work effort is limited to the subcontracting or out-sourcing of final inspection and testing activities.

  For use by prime contractor management when out-sourcing of inspection or testing to subtier vendors is planned.

- ISO 9004, Quality Management and Quality System Elements - Guidelines

  This guideline is used by management to establish a quality management system and to help evaluate the structure and operation of suppliers' quality management systems.

Introduction To the Supporting ISO Family Members:

- ISO 8402, Quality - Vocabulary

  Provides a dictionary of general quality terminology.

- ISO/DIS 9000-3, Quality Management and Quality Assurance Standards - Part 3: Guidelines for the Application of ISO 9001 to the Development, Supply and Maintenance of Software

  A Draft International Standard providing guidance for the interpretation of ISO 9001 for the development, supply, and maintenance of software-based products.

- ISO/DIS 9004-2, Quality Management and Quality System Elements - Part 2: Guidelines for Services:

  A Draft International Standard providing guidance for the interpretation of ISO 9001 for the development and delivery of services.

- ISO/DIS 10012-1, Quality Assurance Requirements for Measuring Equipment -Part 1: Management of Measuring Equipment

  A Draft International Standard providing the requirements for the establishment and maintenance of a calibration system for measuring equipment.

- ISO 10011 Series, Guidelines for Auditing Quality Systems

  Provides guidance for the auditing of established quality management systems.

<u>Using the ISO standards as acquisition assurance tools in the procurement of training products and services:</u>

It is necessary to dispel a myth about ISO 9000. ISO 9001 is a standard defining the minimum requirements for a quality management system, it is not designed or intended to replace all other product specifications used to define technical requirements. As an example, if the operational needs of a simulator require it to be designed and manufactured to the military specification MIL-T-23991 (General Specification For Training Devices Military) then the decision to use this specification is independent of the decision for invoking an appropriate quality assurance standard (e.g., MIL-Q-9856, ISO 9001).

From a management perspective, an argument could be made demonstrating the advantages of using a single standard for defining quality assurance programs requirements versus the use of several procurement-unique standards. Pragmatically, the use of a single standard would reduce the impact on management of the industry's tendency of each element designing, implementing, and reporting on quality issues in their own unique ways. A common denominator to quality assurance would provide management with a tool to integrate the measures of quality across the training acquisition spectrum. This improved reporting capability would clearly support the goal of an effective acquisition process.

No single management tool can be presented as a panacea to all acquisition problems, but each tool selected should effectively and efficiently support the goals of management. Individuals critical of a "centralized" approach to quality assurance may argue that such an approach would limit the ability to tailor or customize quality assurance to user-unique needs. The only aspect of centralization that ISO 9001 would require is the need for reporting status-significant information to management. Each unique community element would have the autonomy to decide how they do business and what measures represent significance to them. Examples of where ISO 9000 standards could be employed within the training community are provided in Table 2.

Table 2. Use Of ISO 9000 Family Of Quality Assurance Standards In Training Systems Acquisition

Acquisition Element	Recommended Application of ISO 9000 Standards
Simulator Hardware and Software Design	1) ISO 9001 2) ISO/DIS 9000-3 (Software)
"Build to Print" NDI/COTS	3) ISO 9002
Training Program Development	1) ISO 9001 2) ISO/DIS 9004-2 (Services)
Training Instruction	1) ISO 9001 2) ISO/DIS 9004-2 (Services)
Maintenance Services	1) ISO 9003 2) ISO/DIS 9004-2 (Services)

<u>Use in Simulator Hardware and Software Procurement:</u>

ISO 9001 could be used as an effective quality assurance substitute for MIL-Q-9858A during the

design, development, manufacture, test, and installation of simulator hardware and software.

Application could also apply to NDI (non-development items) and COTS (commercial off-the-shelf) items in lieu of best commercial practice. If circumstances warrant, ISO could be complimented by other process controls such as DOD-STD-2167A (Defense System Software Development) or specific contract language for inspection and test requirements.

Use In Courseware Procurement:

Courseware, a unique product from the other training system elements, could be developed as part of a training system under the requirements of MIL-STD-1379, and use ISO 9001 as a basis for defining the quality control plan required by that standard. Certainly the tailoring and application of ISO 9001 to courseware development would be different than its implementation for simulator hardware and software. Therefore, ISO 9001 could provide key management information on the status of the development process of this at-times intangible product.

Use In Contract Aircrew and Maintenance Training Procurement:

Arguably, this element of the training community is staffed by some of the most experienced and professional resources within the community itself. Within this sector a direct correlation exists between student proficiency and life and death. The track record is good, the quality assurance standards (under MIL-STD-1379) often unique to specific services and commands. Without being overly intrusive, ISO 9001 could provide government and contractor managements with key indices of performance which are now often reserved for local operational levels.

Use in Contract Maintenance Services Procurement:

When maintenance services transitioned from government to contractor responsibility, the quality assurance practices conducted by contractors were largely representative of governmental practices. As contractor maintenance evolves and innovations such as contractor logistic support shifts additional responsibilities to the contractor, contractors will look to new quality assurance methods for assuring service integrity as well as operational profitability. ISO 9003 could provide the basis for the minimum considerations for quality assurance.

Summary and Conclusions:

As declared in the introduction, the training community's procurement needs are unique due to the diverse scope they entail and the technologies they span. Currently, several quality assurance standards are available and in use throughout the training community. None of these standards cover all of the training system quality assurance needs. ISO creates the opportunity for management to integrate all training system quality assurance elements under a single contractual standard.

The ISO 9000 family of quality assurance standards are comprehensive and tailorable to meet the acquisition assurance needs of all training system elements. Although not currently approved by the Federal Acquisition Regulations for Department of Defense (DoD) use, the ISO standards are worthy of review and analysis for potential future use. Is there management information that a standard such as ISO could cause to become visible and reportable, thereby improving the acquisition process?

Other considerations for the use of ISO may be its international acceptance and use. Procurements involving the participation of foreign governments or contractors may be facilitated by ISO 9000. Not only have many foreign governments adopted ISO 9000, but the Ministry of Defense in the United Kingdom has slated it as the replacement for their MIL-Q equivalents, AQAP-1 (hardware) and AQAP-13 (software).

If commercial considerations continue to a play a role in shaping DoD procurement policy, ISO has become the common denominator for quality assurance standards in the global economy. This preeminent position certainly makes it worthy of consideration for review and analysis. ISO certainly would serve as a minimum benchmark for management in the pursuit of world class performance and quality. If the veteran MIL-Q needs a replacement, ISO 9001 should be considered as a viable replacement.

As all industries introspectively seek new ways at becoming more efficient and effective at managing dwindling resources, quality related processes are

looked upon for assistance. A standard such as ISO 9000 provides the basis for documenting and evaluating many critical management processes within a business and industry. This process can form the analytical basis for total quality management and continuous improvement.

ISO 9000 can provide the basis for total quality management and continuous improvement. ISO 9000 is truly global in dimension and use, provides an integrated quality management approach, and improves management control of acquisitions.

ISO 9000 is worthy of the training Community's consideration and evaluation.

References

1. Tierney, T. "Commercial Quality Standards For Training System Subcontractors And Vendors", I/ITSC Proceedings 8th Annual, Nov. 18 - 20 1986, Volume 1, page 224.

2. Byrnes, D. "Exploring the World of ISO 9000", Quality Magazine, October 1992, page 19.

3. International Organization For Standardization, "Catalogue" 1993.

ENABLING COST-EFFECTIVE DECISIONS ON THE PROCUREMENT OF TRAINING EQUIPMENT BY THE USE OF TRAINING NEEDS ANALYSIS AS A MANAGEMENT TOOL

Lieutenant Commander David Shields Royal Navy
Royal Naval School of Educational and Training Technology
HMS NELSON, Portsmouth, England

Commander Bob Duke Royal Navy
Ministry of Defence, Directorate of Naval Education and Training Support.
London, England

ABSTRACT

The cost-effectiveness of certain items of training equipment hardware has been called into question. It was apparent that no effective means of identifying training needs or selecting media alternatives was included in the Royal Navy's procurement process. This paper describes the processes that have been embedded into the current system to ensure that proper value for money decisions can be made.

ABOUT THE AUTHORS

David Shields is currently the Training Design Officer in the Royal Naval School of Educational and Training Technology (RNSETT), at HMS NELSON in Portsmouth, England. This organization provides training consultancy services to Naval establishments and he has particular responsibility towards the interpretation and application of the Royal Navy's approach to Instructional Systems Development. He has held several training officer posts in the Royal Navy and joined the RNSETT in 1990 after secondment for post-graduate training. He holds a Bachelor of Education degree in the Organization and Administration of Education from King Alfred's College, Winchester and an MA in Adult and Post-Compulsory Education from the University of Southampton, where he is currently registered as a PhD student researching aspects of training and organizational change.

Robin (Bob) Duke is the Royal Navy's Training Equipment Policy Advisor, providing a link between the Royal Navy and Industry. He joined the Royal Navy in 1965 as an Engineering Apprentice and was promoted to Sub-Lieutenant in the Weapon Engineering Specialization in 1974. His recent posts include HMS EDINBURGH as the Weapon Engineering Officer, and the staff of Commander-in-Chief Fleet. He graduated from the NATO Defense College Rome, Italy in February 1992, before taking up his current post in April 1992.

ENABLING COST-EFFECTIVE DECISIONS ON THE PROCUREMENT OF TRAINING EQUIPMENT BY THE USE OF TRAINING NEEDS ANALYSIS AS A MANAGEMENT TOOL

Lieutenant Commander David Shields Royal Navy
Royal Naval School of Educational and Training Technology
HMS NELSON, Portsmouth, England

Commander Bob Duke Royal Navy
Ministry of Defence, Directorate of Naval Education and Training Support.
London, England

BACKGROUND

The Policy of the Royal Navy towards managing the learning needs of its people ensures its 50,000 or so officers and ratings receive training, development and education appropriate to the jobs that they are required to do in the posts that they occupy. Almost all of the instruction is carried out within specialist shore-training establishments. The processes that determine the syllabus ingredients of courses that are primarily education or development are acknowledged to be subjective. The topics instructed in these courses are broadly vocational and the content is aligned wherever possible with the requirements of outside authorities who accredit the courses for the award of civilian qualifications. The Royal Navy has always accepted that the resources devoted to instructing these two activities may not produce a quantifiable pay-off in every individual, but the continued provision of education and development adds much in various non-specific ways that combine to produce high calibre individuals.

It was slightly more than twenty years ago that the Royal Navy became alerted to the potential advantages in increased efficiency and cost-effectiveness that could result from marginalizing subjectivity in the design of 'training', as distinct from education and development, by adopting a systematic approach. The model that was developed drew heavily upon the patterns established by the United States and Canadian armed forces. The building blocks of training technology theory, suitably augmented by practical advice from North American colleagues, provided the foundations of a basic model implemented in 1971. The initial model has evolved somewhat over the intervening years, but the Royal Navy's current system remains comparatively simple and without most of the currently fashionable refinements. The trajectory of changes to the Royal Navy's systems model has been towards simplifying the processes and reducing paperwork rather than towards elaboration.

To illustrate the relative scale of applied training technology processes in the Royal Navy; if the range of activities currently defined by contemporary training technology writing as being essential in the development of training courses were likened to a construction kit, then by assembling it a skyscraper would be erected. By comparison, within the analogy, the activities specified in the Royal Navy's systems model would build a sort of modern thatched cottage; an edifice that will carry out much the same job as a skyscraper but on a smaller scale, with a lower profile and with such retained elements of rustic charm that enable it to pass without remark in a historic village. In the Royal Navy's systems model there is an elementary framework of basic training technology processes. The system's growth has been restricted in line with what is sustainable by a modern, but medium-sized Navy, and adapted to blend with the organizational culture and mores of the Service

BASIC ELEMENTS IN THE RN TRAINING SYSTEM MODEL

The cornerstone of the Royal Navy's systematic approach to training is the process of Job Analysis, the familiar routine of dissolving a job into duty and task

components. The main headings deduced from this analysis are listed to form part of an Operational Performance Statement (OPS) describing Military Capability. Each of the elements in the OPS has a standard NATO 'Training Category' number ranging from 1 through to 5 assigned to it. These numbers indicate the proposed ratio of shore training to on-the-job training (OJT) in each case. A task which has a category 1 against it shows that personnel graduating from the proposed shore-training course will require no formal on-the-job training, whereas a category 5 indicates that full training will have to be undertaken by ships' staff in the operational environment.

The decisions about training categories have far reaching implications for the Royal Navy's separate training and operational commands. A category 1 decision almost certainly means creating high fidelity training situations ashore, and this has clear resource implications for the training commands. On the other hand a category 5 decision imposes a significant training load on ships at sea and this can form an unwelcome burden to busy operational units. By allocating these training categories early in the evolution of a training course design negotiations can take place between the two commands. By the end of this discussion phase accommodations are reached which target shore-training resources towards providing training equipment to tasks that have an agreed low number training category whilst high number categories are agreed for those tasks that can be trained at sea with the least penalty.

Once agreement of the training categories is reached a Training Performance Statement (TPS) is compiled which consists of a list of formal training objectives. This list is then amplified and elaborated into a subordinate level of documentation, the Instructional Specification (I.Spec) which detail the precise instructional activity to be applied. Every training course run by the Royal Navy has six stages of documentation supporting it and these are produced by training design teams who are all uniformed personnel. Each part of the documentation has a specific purpose in the management of up-and-running courses and the goal of the training designers is to precisely define the limits of training ashore and the extent of OJT at sea. The documentation also acts as the communication medium between, the training establishments, who are charged with the design and delivery of shore-training, the training command, who allocates the resources between the various strands of shore-based training, the operational command, who employs the graduates of shore training courses as well as providing the vital OJT element, and, the Ministry of Defence.

OUTLINE OF THE PROBLEM

The limitations to the processes that can be addressed by the Royal Navy's training system occur partly because of the size of the service, but also because of the deliberate policy of using Subject Matter Experts (SMEs) as training designers. There is a paucity of training technology specialists in the Royal Navy and these individuals are used mainly to train SMEs in the techniques of the Service's systems approach to training (SAT) method. The SMEs then spend an average of only 20 months doing training design before they return to operational units and resume their specialist duties. If additional or more complex training design processes were in place then the extra time required to train SMEs in these techniques would be uneconomic when related to the period that they spend doing this job. Because of the shortage of Training Technology expertise there has been little research and development aimed at modifying and refining the Instructional Systems Development (ISD) processes. Thus the Royal Navy's SAT has not adopted a formal process of Training Needs Analysis (TNA) or any systematic method of evaluating training media alternatives. Research into techniques of TNA and the setting up of an in-house systematic process of media selection such as that described by Hougland & Duke (1990) that are carried out for the US armed forces are beyond the resources of the Royal Navy.

Without a formal process of TNA embedded in the Royal Navy's SAT model and with the absence of any systematic examination of Training Media Alternatives (TMA), the range of training and the selection of items of media caused by the introduction into service of new items of ship-borne equipment has placed the RN somewhat at the mercy of the manufacturers. Traditionally, the Warship Platform Manager or Equipment Project Manager has viewed the procurement of

Training Technology (as opposed to Information Technology) as very much a secondary task when compared to the procurement of actual operational equipment. Over the years a predominant philosophy emerged, to procure enough operational equipments to fulfil the needs of the Fleet, and two additional fully operational equipments for operator and maintainer training. Other items of media that may have provided an equally effective or even better platform for training were never considered. This general situation has caused a number of problems:

a. The complexity of Operational Equipment has led to the training facilities lagging behind the introduction of new equipments.

b. Training Equipment, when accounted for in terms of Unit Production Costs (UPC), takes up a larger proportion of the total budget, as the volume of operational equipment to be procured falls.

c. Project Managers have a tendency to treat the training equipment as their contingency margin, and should the project start to go into cost over-runs, which they inevitably do, savings are first achieved by deleting the training equipment.

d. In many instances, the provision of operational equipment, at considerable cost, is not appropriate to meet the training need.

e. Instances have occurred where the operational equipment has been left on the shelf in the store because either there was no money designated for the installation of the equipment, or there was a breakdown in communication between the Project Manager and the training establishment such that the establishment was not aware that the equipment was available.

Even in cases where alternatives to using 'the actual equipment' for training were considered this aspect of training equipment provision has been largely left in the hands of the manufacturers. The contracts have placed no lien on data used by manufacturers to determine what training equipment to supply and the evidence produced by companies to support their proposals have been confined to a 'sales-pitch' for their preferred solution. Various small scale evaluations of the effectiveness of certain items of training equipment that have been installed has revealed that a significant number of items either fell dramatically short of expectations or consumed resources way out of proportion to their usefulness. Two prime examples are:

1. A sonar set that is being fitted into just two vessels. A training unit has been provided by the equipment manufacturers at a cost of £250k ($350k). An examination of the learning outcomes for trainees using the unit has revealed that a Computer Based Training system could have been provided at about £100k ($140k) which would have satisfied the same training objectives.

2. A maintenance trainer for a particular class of vessel is being installed in a training establishment at a cost of some £11m ($15m). However not only are these vessels already at sea and the training is therefore late, but this vast investment has been made to achieve a training load of only 24 people a year.

The cases outlined above are not isolated ones, there are other unhappy instances where equipment purchased for training has fallen well short of its billing. The Navy's requirement for training cannot easily be spelt out in detail at the early stages in the equipment procurement cycle in terms of functional specifications. Therefore in most cases there was a presumption that operator and maintainer training would be carried out using items of actual equipment as the platform. The justification for this position is usually couched in terms related to perceived advantages in creating conditions of verisimilitude. Whilst this assumption is a compelling one, especially for the training of operators, the use of actual equipment incurs some significant penalties when employed as the principal vehicle for training maintainers.

As modern combat weapons systems or radio communications equipment has become more technologically advanced so it has proved to be less useful as training equipment. When first line maintenance of equipment was at the component replacement level it was relatively straightforward for an instructor to create a realistic preconceived fault condition by inserting dud components into a system for

the purposes of training. The learning objectives that were centred on this equipment were able to achieve standards close to those required by maintainers working in the operational environment. Experience in using modern equipment has revealed that the fault finding and rectification elements of these training courses were difficult to achieve in the same manner. The realization that modern equipment had severe limitations when used for training purposes did not permeate to the procurement agencies sufficiently rapidly and items of actual equipment were delivered and installed 'ready for training' only to discover that they were unable to meet the learning objectives created for the preceding generation of equipment. The pattern of instruction in such cases was altered as far as possible to compensate for this. This resulted in a shift of emphasis from using the actual equipment as the principal vehicle of training to one where it was being used as a mere supplement to classroom based theory instruction. This situation created an increase in the costs of shore-training but with a decrease in its efficiency. The unwelcome result of this was a shift of more training to sea, and a concomitant reduction in operational effectiveness.

RE-GAINING CONTROL

Stemming from the current fiscal climate, the principal philosophy guiding statements of policy on the general issue of the Royal Navy's Training Strategy is that training should be carried out where and when it is most cost effective to do so. A separate supporting Training Equipment Strategy has recently been endorsed, the main elements of which are:

a. A TNA, incorporating an Investment Appraisal and Training Media Analysis is now a mandatory requirement before any Training Technology procurement action.

b. The UK's Human Factors Initiative, which equates to the US MANPRINT, will be fully supported, primarily to increase the commonality of equipment man machine interfaces (MMi) and hence to reduce the training requirement and also to increase the range of training that can be conducted in a single facility.

c. The procurement of generic based Training Technology should be the preferred option to meet future requirements, where it can be shown to be cost-effective.

The introduction of these elements into the contractual arrangements for the supply of training equipment is intended to tighten up control on resource exnenditure. Previous attempts at achieving this by tying down precisely the specifications of training equipment with detailed performance criteria had met with little success. This was largely due to problems in defining exact and enduring requirements at such an early stage in the procurement cycle. This process has been likened to 'trying to pin a flag to a moving mast'. At the focus of the Royal Navy's actions aimed at countering the dysfunctionary trends is a set of milestone deliverables. These are listed under the umbrella term 'Training Needs Analysis' and are superimposed on the existing Ministry of Defence (MoD) equipment procurement cycle for capital expenditure equipment projects (see Figure 1). In deciding the ingredients for the TNA it was considered important to ensure that manufacturers carrying out this work on behalf of the Navy follow a systematic approach themselves and that their processes are fully revealed.

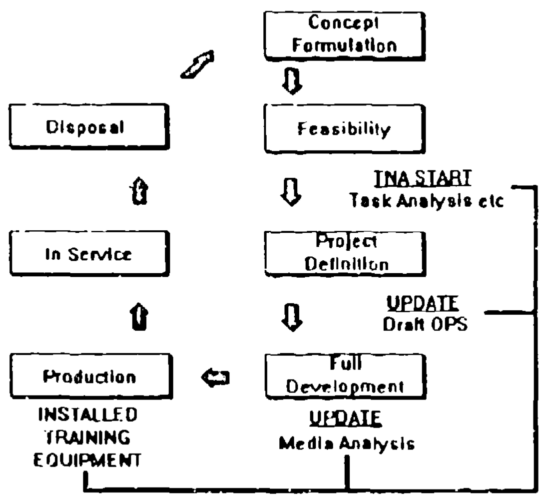


Figure 1 The MOD Procurement Cycle

ENHANCEMENTS TO THE CURRENT PROCUREMENT PROCESS

Deliverable #1

The initial stage in the TNA is carried out as proposals on the feasibility of new systems are being firmed up. As soon as sufficient data is available the first set of deliverable items is produced. The first of these is a full list of duties and tasks detailing what individuals will be required to do in relation to the procedures, operation, repair, etc. resulting from new equipment. This document should be supported by a description of the methodology used to arrive at the job information, and separate work-sheets detailing why, how and how well each task is to be carried out.

Deliverable #2

Once the procurement project has been fully defined the duty task inventory is updated and the second deliverable, the Operational Performance Statement (OPS) is produced complete with agreed outline training categories. This document enables an estimate of the ratio of shore to sea training for each task to be made at a much earlier stage to what was previously available.

Deliverable #3

Once the procurement cycle has moved into the full production phase of the new system the next stage of the TNA is enacted. This action requires there to be a comparison of the OPS to an inventory of existing skills to generate a statement of the additional tasks requiring training. The conclusions reported here need to be fully supported by a description of the evaluation process used.

Deliverable #4

Once the skills to be trained and the types of training categories have been defined the TNA requires that an appropriate and detailed analysis of the media alternatives is undertaken. The results of this should reveal a range of media items that can be employed towards achieving the training goal. A typical report after this stage may present media solutions consisting of different mixes of synthetic training and operational equipments.

Once these four deliverables have been assembled then the Navy is able to evaluate the resource implications for each proffered media solution. At this point trade-offs can be made of on-the-job-training (OJT) load against resource expenditure for training equipment purchases. Once a media solution has been decided upon then the supplier of the media items is locked into providing an installed training system to meet the agreed training categories. This ensures that any media solution develops in line with the modifications made to operational units so as to preserve the OJT/shore training ratio. Because the format of the deliverable items is of a standard type, the data is of immediate use to the Service's own training design personnel. Also, should the Navy decide to adopt its own strategy for the provision of training media then the work done by outside agencies can be easily interpreted into the Royal Navy's own ISD process.

CONCLUDING REMARKS

Problems associated with the detailed control of expenditure in defence contracts have emerged in many countries and in most branches of their various armed forces. In this instance an attempt is being made by the Royal Navy to bring a significant item of expenditure under full control. The processes described in this paper contain nothing new in themselves but the application of such a rigorous system to the procurement cycle for equipment purchases and at such an early stage represents a significant innovation for this organization. In this initiative the Royal Navy is employing the basic concept of TNA as a management tool rather than the more usual interpretation as a set of processes performed by in-house training developers. By using this tool, the Navy is able to lock media selection to the outcomes of training and this represents a fundamental shift of emphasis away from the current obsessive focus on technical data for functional specifications. The Service has been criticised in the past (e.g. McIntosh, 1990) for retaining elements of its cultural heritage at the forefront of current management practices. The measures outlined in this paper illustrate to some extent the Royal Navy's willingness to embrace new techniques whenever they offer the prospect of improved efficiency or assist in addressing dysfunctionary trends.

REFERENCES

Hougland, E. S. & Duke, D. S. (1990). A Process to Evaluate Training Media Alternatives, in Proceedings of the 12th Interservice/Industry Training Systems Conference, November 6-8 1990, National

Security Industrial Association, USA, pp41-52

McIntosh, M. (1990). Managing Britain's Defence. Macmillan Academic and Professional Ltd., Basingstoke, Hampshire.

Ministry of Defence D/DGNMT 10/2/52, (1992). BR8420 The Management of the Royal Naval Training System.

Ministry of Defence, Directorate of Operational Requirements (Sea), 1993. Royal Navy Training Equipment Strategy Paper

Ministry of Defence Public Relations (Central), 1990. The Ministry of Defence 'Blue Book'. HMSO

U.S. AIR FORCE HELICOPTER TRAINING HIGH FIDELITY SIMULATION PROVIDING ADVANCED MISSION TRAINING AND REHEARSAL

Lt Col Edward T. Reed
542d Crew Training Wing
United States Air Force
Kirtland AFB, NM

ABSTRACT

The majority of helicopters in the US Air Force support two distinct and important missions; Air Rescue and Special Operations. Since 1987 the Air Force has been transitioning from a mixed fleet of various models of H-3's, H-53's, and ten UH-60A's to a standardized force of 41 MH-53J PAVE LOWs and 101 M/HH-60G PAVE HAWKs. Combat proven in Operations JUST CAUSE and DESERT STORM; these aircraft, and the crews that fly them, continue to prove that though small in numbers compared to the other services, the Air Force helicopter force continues to be a leader in advanced avionics and combat tactics. As part of significantly upgrading its helicopter force, the Air Force has invested in a comprehensive and fully integrated training and mission support system for the 542d Crew Training Wing (CTW) at Kirtland Air Force Base, NM. This advanced training system combines self-paced computer based training (CBT), electronic classrooms with computer-aided podiums (CAPs) for academic training, Part Task Trainers, and fully networked Operational Flight and Weapon System Trainers providing sophisticated flight simulation capabilities. All systems are controlled by a computer based training management system (TMS). This Air Force helicopter training system provides exceptionally realistic flight training, combat mission training, and mission rehearsal. This paper describes the development, procurement and specific capabilities of the helicopter training and mission support system at the 542 CTW. Also discussed will be lessons learned and future upgrades.

ABOUT THE AUTHOR

Lt Col Edward T. Reed is Director, Aircrew Training Systems, 542d Crew Training Wing, Kirtland Air Force Base, New Mexico. He is responsible for the management and operations of both fixed and rotary wing aircrew training devices which support the wing's special operations and air rescue training missions. He has been qualified in a variety of H-53 variants including HH-53B, HH-53C, HH-53C (NRS), MH-53H, and MH-53J, serving in both Combat Rescue and Special Operations Squadrons. His previous duties have included: Special operations program manager at HQ MAC, Scott AFB IL, HQ USAF, and the Assistant Secretary of the Air Force for Acquisition Staff Washington, DC, Chief Combat Tactics 1 SOW, Hurlburt Fld Fl; MH-53H instructor Pilot, 20 SOS, Hurlburt Fld Fl; and HH-53C Aircraft Commander, 41 ARRS, McClellan AFB, CA. Lt Col Reed holds a B.A. in Political Science from the Citadel, and a Masters in Public Administration from Golden Gate University.

U.S. AIR FORCE HELICOPTER TRAINING

HIGH FIDELITY SIMULATION PROVIDING

ADVANCED MISSION TRAINING AND

REHEARSAL

Lt Col Edward T. Reed
542d Crew Training Wing (CTW)
Kirtland AFB NM

INTRODUCTION

The U.S. Air Force has operated helicopters successfully since becoming a separate service in 1947. Today it continues to employ helicopters in a variety of roles, with the majority of its fleet dedicated to two primary missions; search and rescue (SAR) and support of joint service Special Operations Forces (SOF). In the mid 1980's two events led to a profound change in the force structure and capabilities of the Air Force helicopter force. These events in turn, led to major changes in the way the Air Force now conducts its combat crew training for the SAR and SOF missions. In 1985, the Air Force cancelled the HH-60A NIGHT HAWK program due to the "affordability" of the program. The original requirement for 243 HH-60D's with advanced avionics including terrain following radar had dwindled to 90 more austerely equipped HH-60A's.

About the same time, congressional interest in improving the capabilities of the Air Force's limited SOF helicopter fleet gained substantial momentum. In 1986 Congress funded the first 10 of an eventual 41 MH-53J PAVE LOW III "ENHANCED" helicopters and in 1987 mandated that the entire Air Force fleet of H-53's would be completely modernized and dedicated to the SOF mission. At the same time the Air Force, with congressional support reprogrammed remaining NIGHT HAWK funding and initiated the procurement and modification of Army UH-60A/L BLACK HAWKs. These greatly enhanced H-60's were designated M/HH-60G PAVE HAWK and 101 were approved and appropriated by the Congress by fiscal year 1993. The pace of these programs was very aggressive and required a streamlined and cost controlled acquisition approach in a bureaucracy usually known for delayed schedules and cost overruns. The first MH-53J was required in 1987, twelve months after program initiation. This "business not as usual" challenge including the simulator requirement, was declined by the Aeronautical Systems Division (ASD) of the then Air Force Systems Command (AFSC). It was deemed as not "doable within the proposed cost and schedule". Instead, the Warner Robins Air Logistics Center (WR-ALC) of the then Air Force Logistics Command (AFLC) accepted the program and produced the first MH-53J by July 1987, only 11 months after contract award. *(NOTE: Both organizations have now merged into a single command, the Air Force Material Command, AFMC)* This was a significant success for the dedicated team of SOF specialists at WR-ALC since the MH-53J avionics included terrain following (TF), terrain avoidance (TA) radar, forward looking infrared (FLIR), and a fully integrated navigation suite including doppler, inertial, and global positioning system (GPS) capability. In fact, the MH-53J was the first aircraft in DOD operational with integrated GPS, a fact that later challenged its simulator designers.

With this success under its belt, the decision was made that WR-ALC would accomplish the bulk of the M/HH-60G PAVE HAWK enhancements while ASD would only procure standard Army UH-60A/L BLACK HAWKs with minor radio, crew seat, and paint changes.

MISSION REQUIREMENTS

Though distinct in their heritage and the forces they support, Air Force SAR and SOF helicopter capabilities and tactics became very similar in nature in the mid 1980's. Driven by the enemy threat and advances in U.S. technology, both missions now fly primarily at night in marginal or even adverse weather using multimode radar, FLIR, advanced navigation including GPS, and night vision goggles (NVGs). Both the large MH-53J and the smaller M/HH-60G helicopters share common 1553B data bus integrated avionics with the major differences being the PAVE LOW's radar has TF/TA capability and not all PAVE HAWKs have FLIRs installed. Sophisticated on-board defensive systems include; Radar Warning Receiver (RWR), infrared countermeasures (IRCM), and Flare/Chaff dispensers, on the MH-53J and M/HH-60G. In addition, the MH-53J also has a missile warning receiver (MWR) and

electronic countermeasures (ECM). This almost "overnight" infusion of high-tech combat helicopters caused a significant change in the methods and systems used to train combat crews by the 1550th Combat Crew Training Wing (now designated the 542d Crew Training Wing) at Kirtland Air Force Base, NM.

No longer would the traditional blackboards, 35mm slide-based academics, and simulators with little or no visual capabilities, meet the rapidly changing training requirements driven by the PAVE LOW and PAVE HAWK programs. With few changes having been made in the Kirtland training syllabus since the mid 1970s, the wings' instructors and training specialists began a process that continues to this day of identifying requirements for new training capabilities and continuing to change the way current systems are employed in the school's curriculum. As the Military Airlift Command (now Air Mobility Command) validated the 1550th's training requirements, only two major questions remained to be answered; 1) how fast could funding be obtained for training devices, and 2) who would attempt to manage a training systems program tied to two very aggressive modification and acquisition programs that required concurrent delivery of the training systems. Since ASD had turned down these programs, MAC and the Air Staff turned to another small but dedicated cell of specialists which became the core unit of a highly specialized SOF simulation and training product team.

1987: PROGRAM METHODOLOGY

To meet the most impossible schedule and technical requirements in 1987 to provide a simulator concurrent with its ongoing MH-53J modification program, WR-ALC requested that the Ogden Air Logistics Center (OO-ALC) at Hill AFB, UT provide MAC and the 1550th with training devices. Before this program, OO-ALC accomplished logistical support functions and minor modifications to simulators procured by the training systems program office (SPO) of ASD. To meet the needs of the SOF mission required no ordinary simulator. The MH-53J Weapon System Trainer (WST) had to be capable of simulating both the sophisticated PAVE LOW avionics and defensive systems and the wide range of mission environments including air refueling with HC-130P/N tankers, shipboard landing, and night low-level flight using radar, FLIR, and NVGs. The correlation between the visual out-the-window scene, the FLIR, and the radar displays had not been implemented with any real success on any simulator operational at that time.

After quickly organizing a SOF simulation product team, OO-ALC conducted a survey of contractors considered capable of producing or modifying such a sophisticated simulator in a schedule considered impossible for the industry. Several contractors declined to participate but one accepted the challenge and in December of 1987, GE Aerospace (now Martin Marietta) was awarded a contract for the MH-53J WST. Instead of starting from "scratch", HQ MAC elected to heavily modify an existing CH-3E Operational Flight Trainer (OFT) at Kirtland. This 1973 vintage, non-visual simulator would be used in an attempt to meet the already impossible schedule. Also, the use of the CH-3E OFT allowed the continued training use of a similar era HH-53C OFT with a VITAL III image generator and night only visual system. Attempting to make the impossible "possible", a small team of subject-matter experts, program management and logistics specialists were assigned to the project for its duration. Air Staff, MAC, OO-ALC, and 1550th CCTW personnel empowered to make decisions "on-site" attended all design and management reviews. This "small team approach" had its roots in the way SOF employs its forces in combat and now proved to be the key element in solving the riddle of successful program management within cost and schedule constraints.

From the beginning, all team members understood that an incremental approach to meeting all SOF and SAR training requirements at the 1550 CCTW was required. Systems would be modified or procured as funding within the aircraft programs became available. Failure to control training systems costs meant the deferral of other training devices or aircraft systems. Likewise, proper cost and program controls would be "rewarded" by additional funding for other MAC validated SOF and SAR training systems. This "carrot and stick" approach was well known to the program team and encouraged creativity while discouraging excessive cost overruns which had plagued previous and now subsequent training system programs.

1988-90: THE MH-53J WEAPON SYSTEM TRAINER (WST) -- "THE BEGINNING"

With a reliable but unsophisticated CH-3E OFT as a baseline, the simulator program team set about building what has become one of the most successful and sophisticated training devices in the world. The "user" requirements were certainly challenging but the OO-ALC product team initiated an unusual program to satisfy them. Based on the unique capabilities incorporated in the MH-53J helicopter, the MAC simulator requirements included:

1) A fully replicated, high fidelity cockpit with positions for two trainee pilots and a flight engineer

2) Two complete instructor operator stations (IOS), one for a pilot instructor and one for the flight engineer instructor

3) Realistic flight characteristics and malfunction simulation

4) Multimode radar simulation including TF/TA, weather avoidance, precision ground mapping, and other modes

5) FLIR simulation

6) Complete simulation of the MH-53Js integrated avionics suite including mission computer, 1553B data buses and doppler, inertial, and GPS tactical navigation systems

7) Conventional navigation systems including VOR, ILS, ADF, and TACAN

8) High fidelity, classified simulation of on-board defensive systems

9) Night Vision goggle compatibility of cockpit lighting, IOS consoles, and visual displays.

10) Extensive visual fields of view for the pilots including forward windows, quarter windows, side windows, and chin windows. Only the small center window was considered "nice to have", but not necessary.

11) A high fidelity visual environment including FLIR and NVG operations allowing day, dusk, and night terrain flight operations between 50 and 5000 feet above ground level. Visual approach and landing simulations had to include "brown out" (blowing dirt) and "white out" (blowing snow) operations.

12) Unique training requirements such as inflight refueling from H/MC-130 tankers and shipboard landings under a variety of conditions.

The original MH-53J WST design attempted to meet all the "users" SOF training requirements. The simulator's initial 1988 design capabilities and components included:

a) A high fidelity cockpit with all displays, gauges, and avionics systems fully functional

b) Two comprehensive NVG compatible IOS consoles with full color displays

c) A completely refurbished motion system with a Fokker digital control loading (DCL) system and improved flight "aero" software.

d) A heavily modified F-16 Digital Radar Landmass Simulator (DRLMS) providing all functions of the aircraft's multimode radar.

e) Complete simulation or stimulation of the aircrafts' tactical and conventional avionics suites including the first operational simulation of GPS in DOD flight simulators.

f) Classified electronic combat training using a heavily modified F-16 Integrated Electronic Warfare Training Device (IEWTD)

g) Day, Dusk, Night, FLIR, and NVG terrain flight operations were to be simulated using a 8 channel Compu-Scene IVA image generator with high fidelity visual displays and a variety of data bases and visual models.

As design reviews and in plant integration proceeded, extensive interface between the SOF simulator product team and the using command members continued. A key limitation became apparent in the design in early 1989. Despite effective use in other DOD simulators, Compu-Scene IVA with its maximum capability of 64 cell texture maps did not appear to meet all the demanding SOF NVG and FLIR low-level flight requirements. In April of 1989 GE proposed to replace the Compu-Scene IVA with the newly developed and much more capable Compu-Scene V image generator. SOF "user" evaluations confirmed that not only would Compu-Scene V provide a greatly improved low-level training environment but its ability to use "real world" photo texture derived from imagery and combined with DMA terrain and cultural features would provide SOF with its first mission rehearsal (MR) capability. In September 1989 OO-ALC awarded an Engineering Change Proposal (ECP) to GE substituting the Compu-Scene V for the IVA. A rapid on site data base generation system (DBGS) was also requested. In response to its "first of its kind" mission rehearsal capabilities, the MH-53J WST was redesignated by the Air Staff as a Weapon System Trainer (WST) and Mission Rehearsal System (MRS).

1990: THE MH-53J WST/MRS -- "INITIAL OPS"

The MH-53J WST/MRS was delivered in May of 1990 to Kirtland AFB and accepted as

ready for training (RFT) in July of 1990, just thirty-one months after contract award. Crew training in the simulator was initiated concurrent with test discrepancy resolution and integration of the IEWTD and its electronic combat environment. Also delivered on-site with the WST/MRS was an expandable Sun workstation based database generation system (DBGS) including a camera system capable of digitizing imagery for image generator use. This system would provide the basis for the Air Force's initial development of high fidelity photo specific data bases for advanced training and mission rehearsal in simulators.

As training began in earnest, crew reaction was mixed. The capabilities of the fully replicated cockpit and the dual instructor stations were praised while the aircraft "feel" was identified as requiring improvement. The "feel" problems were quickly resolved by fine tuning the "aero package" using the digital control loading and a highly experienced H-53 pilot with an engineering background. Most disturbing was crew complaints about the out-the-window visual scenes. The primary data base was designated as "Lake Mead". Previously used by other simulators, the Lake Mead data base consisted of 18,087 square nautical miles (SQ NM) of land mass located in southwest Nevada interconnected with a "generic" ocean and islands. This database had been converted from Compu-Scene IVA to Compu-Scene V and this was discovered to be the cause of crew complaints. Key problems with this visual database included a lack of adequate terrain and feature densities. Other problems included its "flat earth" partly "generic" design which affected a variety of training elements. Maps and film strips for the onboard projected map display were unusable in the generic portions of the database. The flat earth design affected the use of accurate lat/long coordinates especially those derived by automated mission planning systems. This affected GPS navigation as well as training realism. The size of the database limited long low-level flight operations so typical of SOF missions. The lack of adequate terrain density in the visual/FLIR database affected the DRLMS generated radar display. In certain types of mountainous terrain the radar display edges resembled fabric cut by pinking shears.

The solution to these problems turned out to be simple and, in the long run, cost effective. Using a database designed with 64 cell texture maps for Compu-Scene IVA just did not take advantage of the more advanced Compu-Scene V capabilities with 256 maps. A decision was made to replace Lake Mead with a geocentric database of New Mexico. This 57,365 SQ NM database was designated as "Kirtland" and included for the first time actual photo-specific landing zones, training areas, and the Kirtland AFB airfield environment. Kirtland became the first database designed to take full advantage of all Compu-Scene V capabilities. In addition, GE provided updated geocentric databases of Nevada ("Indian Springs") and Southern California ("Malibu"). Operational in 1991, these visual databases and their back-transformed radar counterparts were met with immediate aircrew acceptance and provided the fully correlated, multi-sensor, high fidelity simulation that is now the standard for training and mission rehearsal at the 542 CTW. The lessons learned from these early challenges confirmed that Columbus was correct in 1492. The earth is round, not flat and todays rapidly expanding use of the virtual environment for training and rehearsal demands databases that match the real world. Flat earth data bases of non-specific or generic "non-earth" areas will not support the modern military needs of the 1990's and most certainly will not provide a mission rehearsal capability. It also proved that the local Kirtland airfield environment turned out to be one of the most demanding challenges faced by an image generator to date. The local area at Kirtland combines a complex photo specific runway environment located next to 10,000 foot plus mountainous terrain that drives high terrain densities and the large urban environment of the city of Albuquerque which drives equally high feature densities. Many lessons have been learned about database development and modification in providing and updating this "local" database. These lessons have been effectively applied in other training and mission rehearsal databases.

As part of the initial beddown of the MH-53J WST/MRS, in July 1990, a comprehensive training system support center (TSSC) was activated at Kirtland. This Sun work station based network was manned by experienced software and hardware engineers "geared" to support all facets of the MH-53J WST/MRS. TSSC responsibilities were extensive but none were more important than concurrency of the simulator with the flight line aircraft. Traditionally a problem in DOD simulation, the TSSC at Kirtland went on to achieve several critical concurrency "victories". The first was

the almost "overnight" upgrade of the simulator's avionics software due to urgent DESERT SHIELD requirements in the fall of 1990. A later but just as important success was the extensive upgrade of the MH-53J WST/MRS with shipboard compatibility, increased gross weight and service life extension modifications in 1993. This major block modification upgrade (BMU) gutted the simulator and included the replacement of the original IOS with the now common on-site touch screen variant. The MH-53J WST/MRS BMU became ready for training in March 1993. At the time only 1 of the wing's 4 MH-53Js had received the same level of modification; This was concurrency success story in "anybody's book". Starting with a small staff of in 1990, the TSSC has grown to over 33 personnel including database and intelligence personnel.

1990-1991: THE EXPANSION BEGINS TOWARDS AN INTEGRATED SYSTEM

To support the advanced capabilities of the wings new training devices an effort began in August of 1990 to convert all MH-53J and M/HH-60G courseware to electronic media more commonly referred to as computer based training (CBT). As part of the contractor logistics support (CLS) contract, GE Aerospace placed McDonnell Douglas Training Systems Inc. (MDTSI) on a subcontract for helicopter courseware conversion. This multi-year effort continues today with over 300 hours of courseware on contract with 230 hours delivered by May 1993. 14 contractor personnel were assigned "on-site" at the 542 CTW including management, instructional system design (ISD), subject matter experts (SME), production specialists, and computer graphic artists. 386 PC/AT computers were selected for self-paced learning center use with systems capable of classified training also being procured. This hardware was selected since it was affordable for operational units to procure for continuation training at the squadron level. The initial procurement of 8 CBT computer systems has expanded to 33 systems for MH-53J and M/HH-60G Training. All courseware was converted as stand-alone modules suitable for in-unit continuation training and then fully integrated into the total academic courseware of the formal school at the 542 CTW. A computer training management system (TMS) was also procured to manage everything from scheduling of simulators, to classrooms, and even tracking of the moon cycle for NVG operations.

Two key elements in the success of this extensive courseware effort were its responsiveness to changes and improvements, and the collocation of the entire contractor courseware effort with the formal training school and the wings' instructors who functioned as government SMEs. On-site activities also allowed continual oversight of the contractor efforts by government civilian quality assurance representatives (QARs). These QARs act as an interface between the wing aircrew SMEs and the contractor, and also ensure full contract compliance. A key example of program responsiveness was the development of the electronic classroom (EC) with a computer aided podium (CAP) for instructor led classroom academic training. The EC-CAP concept was implemented in 1992 based on student critiques on CBT courseware. Though supportive of self-paced computer training in the learning center, many students felt it had been taken too far. As a result of this trend toward's "electronic baby sitting", a decision was made to incorporate electronic media into the classroom using the same courseware and hardware designed for the learning center but "customized" for instructor led seminars. 386 PC/AT computers were combined with new state of the art computer controlled VCRs called PC-VCRs in a single custom podium allowing complete instructor led multi-media training at costs significantly less than comparable videodisc based systems. Thirty-five inch big screen video monitors were integrated with the CAPs and seven electronic classrooms were quickly procured and installed. These met with immediate student and instructor approval. One final accomplishment of the MH-53J and M/HH-60G courseware at the 542 CTW is its full compliance with the joint service MIL-STD-1379D. This means that any service can use Air Force helicopter courseware without paying to develop it again through another contractor. This will result in substantial savings for the government in the future as programs voluntarily adapt, or are forced to implement this important interactive courseware (ICW) standard. Many of the courseware modules at the 542 CTW are generic such as night vision goggles (NVGs), GPS, and threats. These can be easily duplicated and provided to other organizations for formal school or in-unit training programs on 386 PC/AT computers which are prevalent throughout DOD.

As funding became available, additional training device requirements transitioned from waiting in the que, to contractual realities to compliment the MH-53J WST/MRS. A contract was awarded in June 1990 to GE Aerospace for the conversion of an existing MH-53H Pave Low Night Navigation Trainer (PLNNT) to a MH-53J Part Task Trainer (PTT) configuration. This substantially improved device became RFT in July 1991 and provides training in navigation systems (including GPS), FLIR, and multimode radar operations. "Heads-down" TF/TA flight is possible in a small local data base using an image generator to create both the FLIR and Radar simulation. The MH-53J PTT was the first training device to be equipped with a new "touch screen" IOS. This IOS uses the ADA software language and provides a responsive, user friendly instructor environment that could be updated or modified easily since there was no keyboard and "hard keys" involved. This IOS developed by SBS Engineering became the standard for 542 CTW simulators and was incorporated in the MH-60G WST, MH-60G OFT, and retrofitted into the MH-53J WST/MRS and HC-130P WST. The common IOS at the 542 CTW, simplifies instructor training, allows instructors to operate different simulators on a daily basis, and reduces support costs by allowing one-time changes or updates which affect multiple devices.

In July of 1990 one of the most aggressive training device contracts ever, was awarded by OO-ALC to GE Aerospace for a MH-60G WST. With H-60s beginning to flow in large numbers from PAVE HAWK production lines to operational units, the lack of a simulator limited the wing's ability to meet the demanding world-wide PAVE HAWK crewmember requirements for pilots and flight engineers. The solution again was simple; get a high fidelity simulator fast. It was the program's implementation that appeared somewhat unbelievable. The MH-60G WST would be an all new simulator acquisition program with capabilities identical to the MH-53J WST/MRS, including an 8-channel Compu-Scene V image generator for both out the window scenes and FLIR, a Harris DRLMS for the Multi-mode radar, and a fully replicated cockpit with all tactical and conventional avionics systems simulated or stimulated. As with the MH-53J WST/MRS all displays would be fully correlated so that the crew would observe the terrain out the window, on the FLIR, and on the radar display at the same size, in the same time, and in the same place. Digital control loading and high fidelity sound systems were also used. Again a small dedicated team was formed and with superb cooperation between the OO-ALC product team, the contractors, and the users at the 542 CTW, one of the most sophisticated training devices in DOD was constructed, delivered, tested, and RFT just 27 months after contract award. This record setting procurement was made possible by streamlined management procedures and a dedicated team effort to avoid "reinvention of the wheel", an all too occurring syndrome plaguing many training device acquisition and development programs. If it worked on the MH-53J WST/MRS it was duplicated, if improvement could be achieved it was constrained to allow later retrofit into the MH-53J WST/MRS.

An example of this forward-fit/retrofit philosophy, was the MH-60G WST's Integrated Electronic Combat Simulation System (IECSS) capability by GE and TRW. This "real world" simulator fully replicates the threat environment including ownship signatures, weapons dynamics, threat engagements, and aircraft countermeasures' effectiveness. Unique is its real time simulation of C3I capabilities and their impact on threat effectiveness. Developed for installation in the MH-60G WST and retrofit into the MH-53J, the IECSS was later selected to upgrade the wing's HC-130P WST in 1993. Flight proven UH-60A and SH-60F software were "lifted" from successful Army and Navy simulator programs and combined with new MH-60G code. The tendency to take this already functioning simulator code and spend years rewriting it into ADA was consciously avoided and later fully justified by the performance of another SOF simulator program which made a decision to go "all ADA" and continues to struggle with software delays, cost overruns, and late deliveries. Another key feature of the MH-60G WST was its ability to interfly with the MH-53J WST on the "SOF-NET" intersimulator network (ISN). For the first time SOF and SAR helicopter crews would be able to fly dissimilar formation and practice multiship tactics in high fidelity simulation. SOF-NET would later provide the basis for even more advanced training and mission rehearsal capabilities with additional simulators added in 1993 and 1994.

As the MH-60G WST was beginning its development and production, requirements to upgrade the existing HH-53C OFT to a TH-53A configuration were funded and put on contract. Cockpit reconfiguration was accomplished by

Air Force simulator maintenance personnel prior to their replacement by contractors. This organic effort is estimated to have saved the government $500,000. Additional modifications included the replacement of the entire motion system by BSC, and the replacement of the visual displays and image generator by GE with high fidelity displays and a Compu-Scene V IG. Visual data bases built for the MH-53J and MH-60G WSTs would "play" free of charge on the upgraded TH-53A OFT. Finally, the TH-53A OFT was added to the SOF-NET ISN providing an additional H-53 simulator on the network for tasks ranging from basic formation flight to mission rehearsal. The networked TH-53A OFT was also designed to function as an aggressor attack helicopter and allows the Air Force to conduct defensive air combat maneuvering (ACM) in advanced training free from previous safety and flying hour cost concerns. After a series of complex modification efforts, the TH-53A OFT became fully RFT in October of 1992.

1991-1993: A SOPHISTICATED AND INTEGRATED TRAINING SYSTEM EVOLVES:

What began to evolve at the 542 CTW was a series of integrated helicopter training capabilities. To accomplish its primary missions of initial, refresher or continuation, and operational aircrew upgrade training the 542 CTW capabilities combine computer-assisted instructor led seminars, self-paced CBT, procedural trainers, PTTs, and the first fully networked high fidelity OFTs and WSTs in DOD. This training system provides powerful "tools" to support other user needs such as testing, accident investigation, tactic's development, including networked combat exercises, and mission rehearsal. In the near future these capabilities will be further expanded by networking the 542 CTW with other DOD simulators and simulations on a secure wide area network. As additional experience was gained with the MH-53J WST/MRS and other systems such as the MH-53J PTT were fielded, more and more emphasis and funding were shifted towards the use of simulation and away from "on-aircraft" training. This commitment to the concept of virtual environment training has resulted in 50 percent or 31 of 62 total MH-53J training flights now being accomplished in simulators. Even more dramatic has been the emphasis in the use of simulation for the majority of the most advanced tactical training that culminates in combat crew qualification. In the final MH-53J advanced night phase, 12 flights are accomplished in simulation with only 3 flights and a final check ride conducted in the actual aircraft. This shift towards combat oriented simulation training provides trainees an environment which cannot be achieved at Kirtland in the actual aircraft including night shipboard operations and a "real world" threat environment which can actually shoot down the aircraft if the trainee does not use proper tactics or countermeasures. Lessons learned from this period were translated into the MH-60G WST and the M/HH-60G training program.

As the true potential of networked simulators for training and rehearsal began to be realized at the 542 CTW, a decision was made to develop a centralized facility to observe and communicate with the simulators on the SOF-NET. This training observation center (TOC) became operational concurrent with the SOF-NET ISN in 1993. The TOC is a 41 seat "theater" from which crewmembers can observe up to eight simulators interflying on the network. Large screen displays can show out-the-window and sensor views while a digital map display shows threat locations and the routes of all simulators on the net. Six role player stations and a training director station allow instructor personnel to communicate and scenario "role play" with simulators on the network. Instructors in the TOC become survivors on the ground, AWACS controllers, or enemy personnel intruding on, or jamming communications. The TOC represents a capability unique to DOD simulation facilities and offers training and rehearsal potential limited only by the imagination of the personnel using it.

Another capability integrated with the helicopter simulators was a commercial quality television recording studio designated the audio visual recording system (AVRS). The AVRS is integrated directly with the simulators and can record the out-the-window scene, the sensor displays, and the radar and missile warning displays. Secure and non-secure communications between simulators can be recorded. In addition cockpit mounted cameras and VCRs record all crew activities and intercom audio. All AVRS video can be displayed in the TOC on displays as large as 60 inches. AVRS videos have proven useful for formal school training, in-unit continuation training, and for recording mission rehearsal activities. Rehearsal videos can be viewed by

crews prior to mission execution or used to brief senior decision makers on mission details.

From its modest initial delivery of four workstations in 1990, the on-site DBGS had become the largest in DOD by early 1993. Nineteen contractor data base modelers and 3 intelligence specialists operate this system supervised by a full time government civilian intelligence specialist. The greatly expanded DBGS uses an extensive and fully integrated computer network of Sun and Silicon Graphics computational systems. The original black and white Eikonix camera was replaced by programmable high speed scanners capable of digitizing everything from black/white and color photographs to all forms of maps, blueprints, videotape, and three dimensional models. Scanning tasks that previously took 30 minutes now take 4 seconds. Imagery processing software then analyzes, enhances, and manipulates this wide variety of imagery and video. This imagery is then combined with standard terrain (DTED), cultural (DFAD) features, and digital chart products such as DCW from the Defense Mapping Agency (DMA) to create correlated geo-centric and photo specific visual and sensor data bases.

The original DBGS II software was replaced with GE's advanced training and rehearsal generation toolkit (TARGET) software. TARGET is now a proven method of rapidly generating high fidelity data bases and can produce data bases and models in a variety of formats including Compu-Scene IVA, V, VI, PT-2000, and Project 2851 SIF formats. This "single" production effort with multiple data base formats has already been proven to dramatically reduce data base generation costs and schedules. Data bases have been built or transformed for Air Force F-16 and C-5B WSTs and additional "customers" are waiting in the que. The capabilities of this advanced DBGS include the ability to update existing data bases with new imagery within 72 hours and can produce extensive photo-specific data bases from "scratch" in five days. Data bases produced by the 542 CTW have proven themselves in a variety of simulators, providing advanced training and rehearsal capabilities which continue to improve and expand.

An example of the types of data bases available for use at the 542 CTW is the southwest data base operational in Fiscal year 1993. A combination of the three previously separate Kirtland, Indian Springs, and Malibu data bases, this 179, 810 SQ NM data base will be one of the largest in DOD. Fully correlated with matching FLIR, radar, and navigation data bases, this represents a new standard in training realism for both rotary and HC-130P simulators. This data base supports locally networked training now, and will support joint exercises in the future using wide area networking.

By 1993 a fully integrated and sophisticated training system had evolved at the 542 CTW for the MH-53J and M/HH-60G helicopters as illustrated by Figure 1. Key aspects of this system are being expanded to include HC-130P/N fixed-wing tanker aircraft training. This system provides high quality training from registration to graduation including classroom, learning center, simulation, and on-aircraft training. All networked simulation can be monitored and evaluated from the TOC. All student functions are controlled by an expanding training management system. A unique aspect of this helicopter training system is the use of contractors only in support roles such as courseware development, data base generation, and training device maintenance. Combat qualified Air Force instructors do the teaching in both simulators and aircraft. In addition, the government, not the contractor, controls the course design and changes. The result is a unique government and contractor team effort which responds quickly and cost effectively to mission training and qualification without the contract changes and schedule delays associated with the more common Air Force "totally contractor operated" aircrew training system. Currently the 542 CTW is leading, in helping the Air Force redefine its concepts of contractor and government training systems including government and contractor duties in complex, multiple aircraft combat crew qualification courses.

1994 and Beyond: The Future

In 1994 M/HH-60G training will incorporate the first of two new MH-60G OFTS. These non-motion but highly realistic training devices are designed to augment the capabilities of the MH-60G WST. Built by SBS Engineering and featuring a new "cross-view" wide display system and a six channel PT-2000 IG, the MH-60G OFT will also allow day, dusk, night, and NVG flight operations in highly realistic data bases. A replicated cockpit will include operational radar, FLIR, and navigation systems. Seat shakers and a digital audio system will prevent simulator sickness and

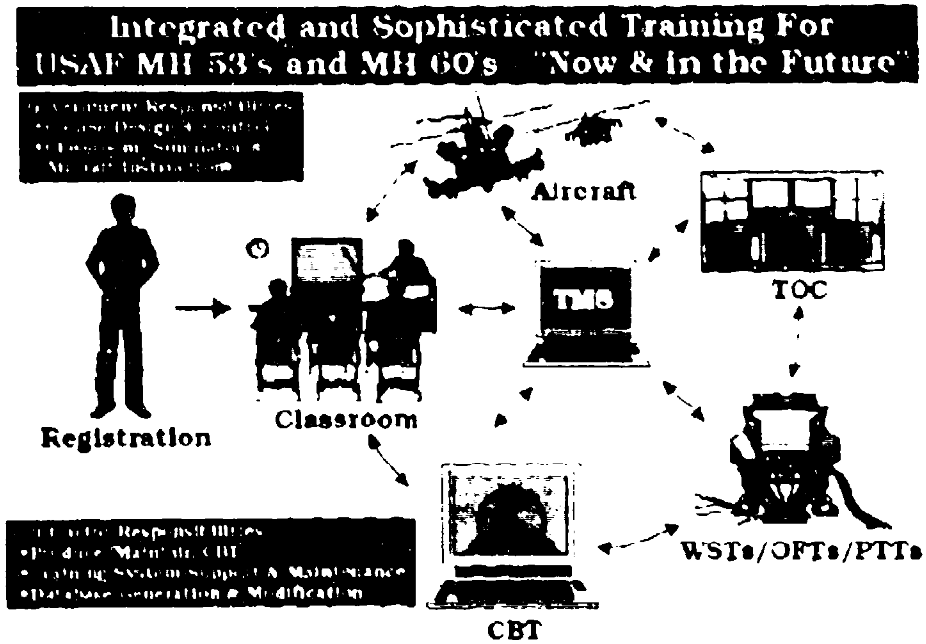


Figure 1

provide realism for basic and combat training tasks including low-level flight and shipboard operations. Plans call for the MH-60G OFTs to be added to the SOF-NET ISN.

Another important 1994 addition to the integrated helicopter training system at the 542 CTW is the Aerial Gunner and Scanner Simulator (AGSS) produced by BSC. The AGSS will train aerial gunners and pararescue crewmembers in NVG scanning techniques and aerial gunnery using 7.62 mm miniguns or .50 caliber machine guns. Computer scored air-to-air, and air-to-ground gunnery against fixed or moving targets will be possible. The AGSS will be capable of independent flight operations using small, low cost, image generated visual data bases. An integrated, networked mode will allow full crew operations between either the MH-53J WST/MRS or the MH-60G WST. In this mode the AGSS will use the TH-53A OFT image generator and will operate on the SOF-NET ISN in a COMPU-SCENE V data base common with the MH-53/60 WSTs.

SUMMARY

This advanced, integrated training system, featuring networked high fidelity simulators started as a single simulator and progressed as technology and funding permitted. With hardware and software today, not briefings and promises for tomorrow, the 542 CTW is now an acknowledged leader in the field of advanced training and simulation. We hope others will share our experiences and learn from them to control costs and prevent "reinvention of the wheel". This is true in the case of the HC-130P WST which will be receiving substantial upgrades in 1993/94 by CAE-LINK to make it common with the MH-53J and MH-60G training systems. A new programmable DRLMS, IECSS, cockpit upgrades, SOF-NET integration and COMPU-SCENE V visuals will result in the most advanced and capable C-130 simulator in the Air Force in 1994\95. The key to the success at the 542 CTW was the incremental approach to developing the systems, users that clearly understood what they wanted, and program management personnel who actually listened to their users needs while structuring programs to meet them in a timely and cost effective manner.

Authors Note:

The author and the 542 CTW would like to extend their most sincere appreciation to the program manager and product team for SOF helicopter training systems (OO-ALC/LIRAC) at Ogden Air Logistics Center, Hill Air Force Base, Utah. Without this dedicated group of acquisition specialists, and our participating contractors, none of what is described in this paper would be possible. They are truly our partners in advancing training technology. Thanks!

PUTTING THE *TRAINING* INTO TRAINING SYSTEM DESIGN: IT DOESN'T HAVE TO HURT

Jolene O. Pike
Tamara L. Busch
Lisa R. Carlson
Hughes Training, Inc.
Minneapolis, MN

ABSTRACT

The recent DoD emphasis on MANPRINT and an integrated approach to system development should be applied not only to defense systems but also to training systems. This emphasis requires a change in the focus of how training system design is accomplished. To ensure maximum training effectiveness, the expertise of training personnel and Systems Approach to Training (SAT) products need to be incorporated into the training system design process.

[illegible]ctive of this paper is to present a management approach which may be used [illegible]to training considerations smoothly into the training system design process. [illegible]oach identifies key SAT training products and methods of information [illegible]nge which may be used, as well as desired areas of training expertise which contribute significantly to training system design. This approach is intentionally modular, giving it the flexibility to be applied to any of the training system design processes practiced in a variety of government and industrial settings. Application of this approach will ensure that critical training issues are addressed early in the design of a training system, making that training system better suited to meet the needs of instructors and students, and ultimately providing a more effective training program

This paper describes the types of training expertise which may be used, the types of SAT training products which may serve as input, and a management structure which will facilitate communication and coordination during training system design efforts. Finally, this paper discusses the benefits of including training personnel and training products in the training system design process.

ABOUT THE AUTHORS

Jolene Pike, Tamara Busch, and Lisa Carlson are all Systems Training Analysts at Hughes Training, Inc. in Minneapolis, MN. Ms. Pike has been responsible for the management, development, and conduct of operator training projects associated with the Joint STARS Ground Station Module (GSM) program. Ms. Busch is currently leading Short Range-Unmanned Aerial Vehicle (SR-UAV) operator training analysis efforts. Ms. Carlson is currently responsible for leading the effort to develop an operator training course for the latest in a series of Joint STARS GSM programs. The authors have all been involved in numerous training efforts, have extensive experience in each phase of instructional system development, and have applied their experience in the training arena to the design and development of training devices and instructional tools. Ms. Pike holds a Bachelor of Arts degree in Psychology from Hamline University. Ms. Busch holds a Master of Arts degree in Industrial Relations from the University of Minnesota and a Bachelor of Arts degree in Psychology and English from Hamline University. Ms. Carlson holds a Bachelor of Arts degree in Psychology and English from Hamline University.

PUTTING THE *TRAINING* INTO TRAINING SYSTEM DESIGN: IT DOESN'T HAVE TO HURT

Jolene O. Pike
Tamara L. Busch
Lisa R. Carlson
Hughes Training, Inc.
Minneapolis, MN

INTRODUCTION

Statement of the Problem

Traditionally, training device design has been performed by the engineering disciplines in isolation from the courseware development process. The activities associated with building a training device have been performed parallel to, but separate from, the activities associated with developing training materials. In part, this lack of coordination has been due to a lack of knowledge about the training processes, products, and expertise that may contribute to training device design. Additionally, there has been no formalized or accepted structure in place to facilitate communication, coordination, or interaction between the engineering and training functions.

With the increasing complexity of current weapon systems, the increased emphasis on learning higher-level cognitive skills (e.g., decision making), the reduction in the force structure, and the increasing sophistication of training systems, the rigid separation between engineering design processes and training development processes is becoming unacceptable. In addition, with the MANPRINT and IMPACTS initiatives, there is now a requirement to consider human performance issues early in the system design process; this requirement exists for both prime system equipment and training devices. To address the MANPRINT requirement, training system device designers must begin to incorporate knowledge, data, and processes from other disciplines such as safety, human factors, and training.

Putting the *training* into training system design means using training expertise, processes, and products to develop training systems which will provide the optimum training environment for both instructors and students. The integration of training into the design process may be facilitated by a greater understanding of what training has to offer and by a management structure which encourages and supports interaction during the training system design process.

Definitions

Before describing what training has to offer or how management can support the interactions between the training and engineering functions, some definitions of training development, engineering design, and training system design are necessary. These definitions will clarify the nature of the basic activities which will benefit from the management structure discussed later in this paper.

<u>Training Development</u>– Training development is a structured and systematic process which is used to produce training courseware. A variety of training development methodologies have been described and utilized by the military services, commercial training companies, and academia. Two highly accepted models for training development are the Instructional Systems Development (ISD) process and the Systems Approach to Training (SAT) methodology. General descriptions of these methodologies will help to provide a basis for understanding the training system design process.

Instructional Systems Development is a proceduralized process used to design and develop effective and efficient training. The ISD methodology is the approved interservice procedure for instructional development. The five phases of the ISD model are: Analyze, Design, Develop, Implement, and Control. Each phase of activity has a set of sequential and interacting steps and the output of each step provides the input needed to accomplish later steps. Later steps also provide feedback to earlier steps.[1]

Each military service provides specific, documented guidance for performing ISD processes. Development involves the actual writing or production of training materials such as outlines, narratives, or scripts. The Systems Approach to Training (SAT) model, described in TRADOC REG 350-7, is used by the Army to develop its training programs. SAT includes analysis, design, development, implementation and evaluation to determine the who, what, where, when, why and how of training.[2] The analysis phase includes the analysis of the mission and the job to determine the specific tasks, knowledge, and skills which require training. The design phase includes converting tasks into learning objectives, sequencing training, preparing course outlines, selecting media, planning for trainee evaluation, constructing written performance tests, and identifying facility and resource requirements. Implementation includes the conduct and management of the validated and approved training program. Evaluation is an on-going process and is conducted to assess the accuracy and effectiveness of the training program.[3]

The training development model described in this paper (see Figure 1), includes a new element—management—which has been added to the traditional ISD training methodology. This addition is supported by the management and planning tasks defined in MIL-STD-1379D.

Engineering Design Activity– There are many formalized systems engineering design models available for use and different industries apply, adapt, or develop their own design models (See Chase; Blanchard & Fabrycky; Hatley-Pirbhai). Some models focus more on requirements definition processes, while other models stress the importance of prototyping and testing. Engineering design typically includes activities such as the identification of design requirements, the development of a conceptual design, a preliminary system design, and a detailed design, design support, development of prototypes, and the transition from design to production.[4]

In our model of training system design, the engineering design activity includes three basic phases of effort: 1) Requirements Definition, 2) Preliminary Design, and 3) Detailed Design. These basic phases were considered to be the common elements in most design models and coincide programmatically with Systems Requirements Reviews (SRR), preliminary design reviews (PDR), and critical design reviews (CDR). Requirements definition includes the identification of system, software, and hardware requirements. Preliminary design includes system concept development, the allocation of hardware and software functions, and the development of a system architecture. Detailed design includes in-depth system design and prototype development, test, and evaluation.

Training System Design– A training system includes both training devices and a training program and may be defined as an integrated combination of all elements necessary to conduct training.[3] A training device may be defined as the hardware and software designed exclusively for training purposes. A training device may use simulation or stimulation to support the learning of concepts, procedures, and complex operational skills. A training program consists of the courseware used to transfer knowledge.

Although the overall training system development process has several phases of activity, including design, development, integration, test, and support, this paper focuses exclusively on the design process. For the purposes of this paper, training system design is comprised of two major activities: engineering design and training development. The design of a training device is accomplished by the engineering activity whereas the creation of the training program is accomplished through application of the training development process.

Figure 1 shows a model of training system design which includes both the training development process and the engineering design activity. A key feature of this model is the interaction occuring between the training and engineering disciplines during design.

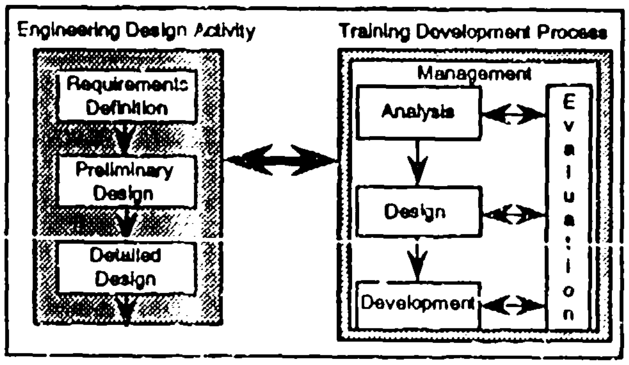


Figure 1 Training System Design Model

Design is the process of planning for the implementation, construction, or production of a training system. The primary output of design is a "blueprint" of the training devices and courseware that will be used as a guide during the development process.

DISCUSSION OF TRAINING PERSONNEL EXPERTISE

There are many different types of training expertise, and many different roles which a training professional may assume during the entire life cycle of a training system. During the period of training system design, six of these roles are particularly useful; they include the training analyst, the instructional designer, the media specialist, the courseware developer/author, the instructor, and the measurement specialist. Although training professionals may draw on additional areas of expertise when supporting training system design, the areas listed above are considered the most beneficial.

Different perceptions may exist as to what types of skills are part of each of these six roles. For clarity, a brief description of each role for the purposes of this paper is included here. There will be further discussion later in this paper on how the expertise from each of these roles supports various aspects of the training system design process.

Training Analysts

A training analyst identifies and describes *what* is to be trained. An analyst develops a list of the trainee tasks, considers the contexts in which the tasks may be performed, and identifies the skills which must be trained to enable the tasks to be performed. The analyst also assesses the characteristics and abilities of the personnel available to carry out those tasks. The training analyst must consider the capabilities of both the candidate trainees, and the candidate instructors.

Instructional Designers

An instructional designer defines and describes *how* the specified tasks are to be trained. The instructional designer must structure objectives that will train the identified skills, and consider how to sequence the objectives for the most effective learning. The instructional designer decides how to present the tasks and related lesson content to the trainee, and also determines how best to evaluate whether the students have accomplished the stated objectives.

Media Specialists

A media specialist focuses on identifying the best methods and media for conveying the necessary lesson content to the trainees. A media specialist is generally familiar with the range of traditional, proven instructional methods, and also has a working knowledge of newer training methods and state-of-the-art media options. It is the media specialist's job to choose, based on the trainee population, the specific training methods and learning principles that will optimize trainee learning.[5]

Courseware Developers/Authors

A courseware developer's main job is to create lesson materials. The developer must have a solid technical understanding of the topic being trained. The developer shapes this subject matter expertise into lesson content, using the guidelines resulting from the efforts of the training analyst, the instructional designer, and the media specialist.

Instructors

An instructor is the leader of a training program session. The instructor's role can range from merely guiding the learning process, to giving all the information directly to the trainees.[6] The instructor may not be involved in the design or development of the lesson materials per se, but he or she must have a thorough understanding of those materials and how to use them effectively, as well as knowledge of the topic being addressed, in order to control trainee learning.

Measurement Specialists

A measurement specialist focuses on the means for evaluating various aspects of training. These aspects range from defining overall evaluation criteria and standards for performance, to identifying the criteria for determining whether particular objectives have been met, to assessing the overall effectiveness of the training as a whole. Measurement specialists also design evaluation instruments, and provide guidance in how to interpret data collected (manually or automatically) for purposes of evaluation.

DESCRIPTION OF TRAINING PROCESSES AND PRODUCTS

The training development process has tangible products which may be incorporated into the training device design process. The discussion which follows provides some examples of the kinds of traditional training activities and products that could be utilized by design engineers. These examples are derived from the analysis and design phases of the ISD model. There are also key training products associated with development and evaluation; these products and their uses are discussed in the management structure section of this paper.

Mission Analysis

The purpose of mission analysis is to develop an overall picture of the context in which a particular job must be performed. Mission analysis involves a review of mission requirements, environments, and events. The main product of a mission analysis is a clear mission description. Design engineers may benefit from mission descriptions by gaining a better understanding of *why* certain training system features and capabilities are necessary. Without some knowledge of the operational context, it is often difficult to discern why certain instructional functions are critical.

Task Analysis

There are many types of task analysis and many more task analysis techniques. Task analyses are conducted to determine what tasks, sub-tasks, and task elements must be trained to ensure successful performance on the job. Collective task analysis, job or duty analysis, and individual task analysis are performed to identify all of the tasks associated with a particular mission, job, or duty position; critical task analysis identifies those specific tasks which must be trained. The products of task analysis include job descriptions and training task lists. The use of job descriptions and task lists by design engineers significantly contributes to an understanding of what students are expected to do and clarifies what aspects of the overall mission and job must be trained. This awareness of what tasks are being trained enables designers to better identify what system functions are necessary to support training.

Person Analysis

The purpose of a person analysis is to develop an accurate description of trainees and/or instructors. A person analysis includes an assessment of the knowledge, skills, abilities, and other characteristics possessed by those individuals who will be trained or who will conduct training. The most significant outputs of a person analysis are a target audience description (TAD) and a prerequisite skills listing. To design a training device that is operable by the target audience, engineers need to understand the skills instructors and students bring with them to the training situation.

Learning Analysis

Learning analysis is the process of identifying the knowledge and skills that must be acquired for a student to achieve mastery of a job, duty, or task. Learning analysis also considers how students may learn best and in what instructional setting certain objectives should be taught. The products of a learning analysis include a listing of knowledge and skills, learning objectives, and instructional setting descriptions. In addition to understanding the mission context, the tasks that must be trained, and the skills people bring with them to the training situation, design engineers need to have an awareness of how student learning occurs. With knowledge of the learning situation, designers are more likely to ensure incorporation of those instructional features which most effectively support learning.

Requirements Analysis

Requirements analysis includes activities such as describing essential training system characteristics and functions, identifying necessary training system modifications, and considering various training system alternatives. The requirements analysis process provides absolutely critical input to the engineering design effort. Training requirements analysis differs from the engineering requirements definition process in that the focus for training is purely on identifying what is necessary to conduct effective training. The engineering requirements definition is more focused on what the system must do from a hardware and software standpoint. The most important output of the training requirements analysis process is a description of training program requirements.

Throughput Analysis

The goal of throughput analysis is to identify the numbers of instructors, training devices, and students required to support the fielding and operation of prime mission equipment. Throughput analysis examines various combinations of resource variables (e.g., course length, percent of hands-on time, crew size, etc.) to determine the optimum mix of resources necessary to provide effective training. Throughput information is important to the engineering design activity because the capabilities and functions of any training system are dependent, in part, on the number of devices available and on the number of people using those devices. For example, a system that must support one instructor managing eight student stations will be much more complex in functionality than a system which requires one instructor to manage only two student stations. If an instructor must manage many devices and the learning experiences of many students, the training system needs to include automated tools to help the instructor monitor and evaluate student performance. To build a useable training system, engineers should understand the significance of throughput and workload issues.

Media Analysis

The main objective of a media analysis is to determine what categories of equipment, technologies, or materials are the most effective means for delivering training. Media analysis includes an assessment of candidate media and the matching of appropriate media to a specific set of learning objectives. For design engineers, the most useful output of a media analysis is a listing of desired media attributes and features. This listing provides guidance as to what kinds of features should be incorporated into the training device. The information provided by a media analysis allows designers to focus their efforts and concentrate on those system capabilities that are necessary and critical for training.

Measurement Analysis

Measurement analysis is usually associated with test development. The purpose of measurement analysis is to identify what should be measured and how it should be measured to best assess a student's grasp of the training objectives. One of the products of a measurement analysis is a performance measurement requirements list. Measurement analysis may contribute to the engineering design activity by identifying those key aspects of student performance that must be monitored, collected, and evaluated.

Fidelity Analysis

The main objective of a fidelity analysis is to determine the minimum levels of training system fidelity required to adequately train critical tasks. Fidelity analysis is performed to determine the levels at which the physical components, functional capabilities, and environmental aspects of an operational system should be replicated in a training system. Fidelity level requirements are the main output of a fidelity analysis. By specifying what capabilities need to be implemented, and at what *level* those capabilities should be implemented, fidelity information provides invaluable guidance to design engineers.

MANAGEMENT STRUCTURE FOR TRAINING SYSTEM DESIGN

The main purpose of the management structure presented in this paper is to provide a practical and flexible approach, or methodology, which may be applied during the design of a training system to ensure training considerations--requirements, jobs, personnel, issues, etc.—are effectively communicated and understood at the critical points in design for integration into the training device. The management structure addresses the types of training products available, the kinds of support capabilities which may be exercised, the appropriate timing for specific training inputs, the means of presentation and recommended utilization of training products, and the expertise provided by training personnel during training system design.

Overview of Management Structure

The management structure is an elaboration of the interaction recommended between the engineering design activity and the training development process during training system design. This interaction was depicted with the bold double-headed arrow in the training system design model (see Figure 1).

The success of the management approach presented in this paper is dependent upon a few important assumptions. First, it is assumed that the

training system design team, which is comprised of a selected group of individuals with various technical backgrounds (i.e., systems, software, hardware), must include at least one training professional.

The second assumption is that activities and efforts which comprise the phases of the engineering design activity and the training development process are understood and practiced. The intention of the management structure is to focus on the insertion of customary training products, processes, and expertise into a collective, general engineering design activity; the management structure does not address what is required to manage either the engineering design activity or the training development process individually.

Finally, it is expected that the recommended interaction process will not be natural or unanimously understood, but instead will be initially awkward and technically challenging; therefore, it is assumed normal management support functions (e.g., coordinating, directing, monitoring) will be necessary to explain the merits/benefits of the management structure, facilitate and encourage data exchanges, and create a positive work environment.

Description of Management Structure

To better understand the management structure, a brief description of the components or elements of the management structure is beneficial. There are two components in the management structure which are essentially fixed (i.e., non-changing) and not modular in nature. The fixed components provide the framework upon which the management structure is built.

One of the fixed components is the overall context in which the management structure resides. The modular components of the management structure are enclosed between the engineering design activity and the training development process presented in the training system design model (see Figure 2). This overall context provides the framework for interactions, as well as the time reference for exchange associated with the management structure by virtue of the parallel phases contained within the training system model. This helps ensure coordinated schedules for training and engineering, which allow interactions and exchanges to happen effectively.

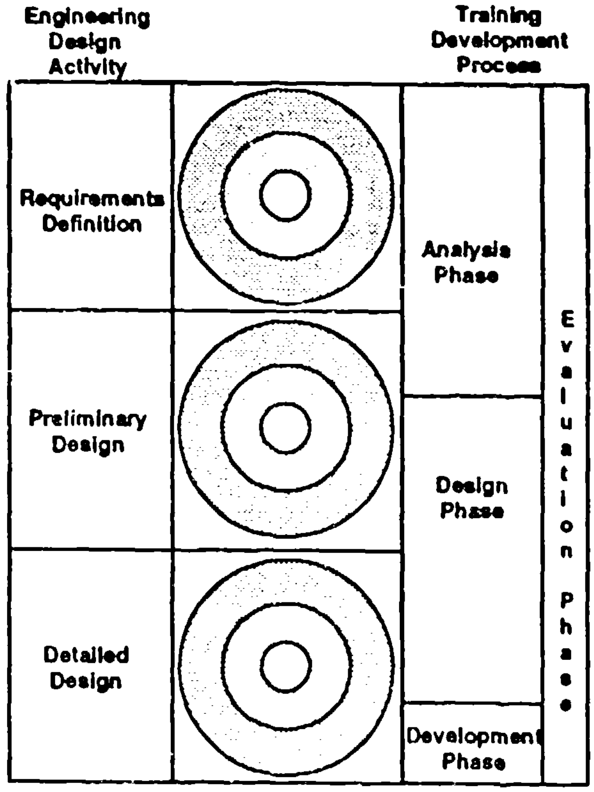


Figure 2 Fixed Components of Management Structure

The requirements definition, preliminary design, and detailed design activities comprise the second fixed component of the management structure. These activities normally occur during training device design. The management structure is built around these core design activities, and focuses on providing additional contributions during these activities that will ultimately enhance the quality of the training device, training program, and eventual training system.

In addition to the fixed components, there are components within the management structure intended to be modular. It is expected, in order to provide the flexibility to apply the management structure to many types of training system design efforts, that each of the modular components and its respective attributes may be used in any combination without jeopardizing the integrity of the overall management structure. The management structure is based on four unique modular components:

1) <u>The Training Products</u> – The training products are referred to as either direct or supporting inputs. The direct inputs are those training products, such as task inventories, job

descriptions, and resource requirements list, which result from the training development analytical processes mentioned earlier in this paper. The supporting inputs are training products, such as test evaluation criteria and sample courseware, which may be created during the development and evaluation phases of the training development process to assist in specific engineering design activities, but that would not normally need to be developed if no such support were required.

The training products are represented in the management structure diagram with boxes (see Figure 3). A box containing a label is provided for each of the training products which may be considered as an input during a particular phase of the engineering design activity.

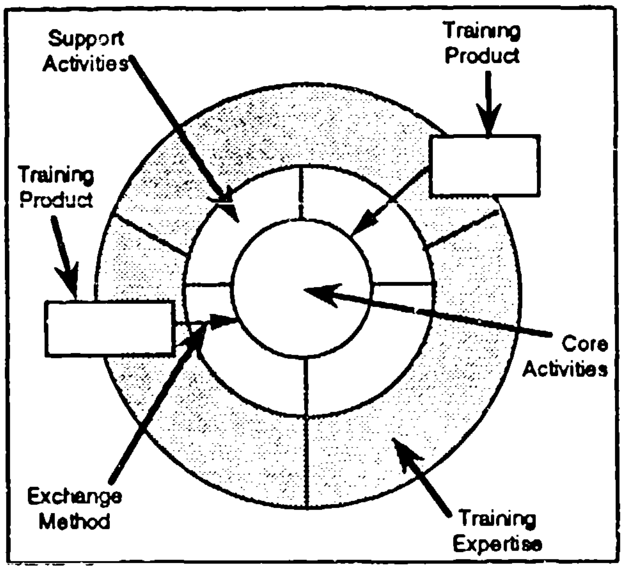


Figure 3 Management Structure Modular Components Key

2) The Support Activities – The additional "perks" or activities which result naturally from the synergy inherent in the integration of training products and personnel into the training system design are referred to as "support activities." For example, during preliminary design activities, training personnel would be available to assist in smart decision making during trade-off studies; therefore, trade-off studies are considered a support activity.

In the management structure diagram, the support activities are represented with the lightly-shaded, inner circle surrounding the core activities circle (see Figure 3). Examples of possible support activities which may enhance the core activities associated with a particular phase of the engineering design activity are listed within the inner circle in the management structure.

3) The Training Expertise – The unique areas of training background and knowledge possessed by training personnel are referred to as the "training expertise." The types of training expertise which are addressed in the management structure were discussed earlier in this paper (i.e., training analyst, instructional designer, instructor, etc.).

The training expertise is represented by the dark-shaded, outer circle which surrounds the support activities and core activities circles (see Figure 3). In the management structure, the specific types of training expertise that complement the core activities associated with a particular phase of the engineering design activity will be listed within the outer circle.

4) The Exchange Method – The method, mechanism, or manner used to provide, disseminate, and share data is referred to as the "method of exchange." The methods which may be used to exchange data are not specified in the management structure. Which attributes of this component to use should be determined by the user of the management structure who knows best which methods will be reasonable and most effective in his or her organization. Some suggested types of exchange methods to consider include formal and informal documentation, formal and informal presentations, group and one-on-one dialogues, formal and informal meetings, brainstorming exercises, question and answer sessions, discussion groups, interviews, points of contact, briefings, and reviews. It is recommended that users of the management structure take advantage of the data exchange opportunities created by formal documentation and formal review requirements.

Although the specific methods to be used to exchange data are not actually listed on the management structure, arrows are shown connecting each training product to the appropriate core activity to represent the need for an exchange at those points (see Figure 3). The arrow also reinforces the relationship of training inputs to core activities.

Presentation of Management Structure

The management structure depicted in Figure 4 shows the relationship of the training system design and the training products, processes and personnel—or components—described and discussed in this paper. The engineering design activity, diagrammed in a chronological manner on the left-hand side of the figure, and the training development process, diagrammed in a chronological manner on the right-hand side of the figure, enclose the management structure. The center of the management structure, which is the interaction between the engineering design activity and the training development process, is divided into three distinct sub-structures to support each of the phases of the engineering design activity. Each sub-structure is modular and contains the possible training products, support activities, and training expertise that are complementary to the core activity it supports.

Using the management structure is as simple as 1) customizing the management structure and 2) incorporating the management structure into the overall management of the training system design effort. Customizing the management structure includes selecting the training products and support activities which are either applicable, available or beneficial, identifying training personnel with the desired training expertise, and determining the appropriate, yet feasible, method of exchange for each training input. Incorporating the management structure into the overall management of the training system design effort involves including the management structure in the everyday management activities of planning, scheduling, directing and monitoring.

The flexibility and the practical aspects of the management structure, along with the natural synthesis which exists within the training system design model and a strong desire to design—and eventually develop—quality training systems, will ensure that training considerations are successfully integrated into the training device.

BENEFITS FROM CONSIDERING TRAINING

In conclusion, there are several benefits to be gained by incorporating training into the engineering design effort. First, integration of activities contributes to a concurrent engineering thrust and demonstrates responsiveness to the MANPRINT and IMPACTS initiatives. Second, integration provides additional focus and direction for the training system design process. This focus is likely to result in a training device design that more closely meets the critical training requirements and that is a "friendly", useable system. Also, the possiblity of re-engineering to retrofit necessary capabilities is reduced. Third, the use of training products eliminates redundant analytic efforts and may result in cost savings.

Finally, the major benefit of applying the management structure described in this paper is a reduction in the "pain" associated with communicating across functional teams. By knowing who training professionals are, what kinds of training products are available, how training information can contribute to training device design, how training may support the design process, and what methods of exchange may be utilized, both engineers and training personnel will feel more comfortable with their roles in the training system design process. Putting the *training* into training system design is necessary, but it doesn't necessarily have to hurt.

SUMMARY

To summarize, this paper has described training expertise and training products which should be integrated into the training device design process. Additionally, this paper has presented a management structure model which may be applied to facilitate the incorporation of training into the design process.

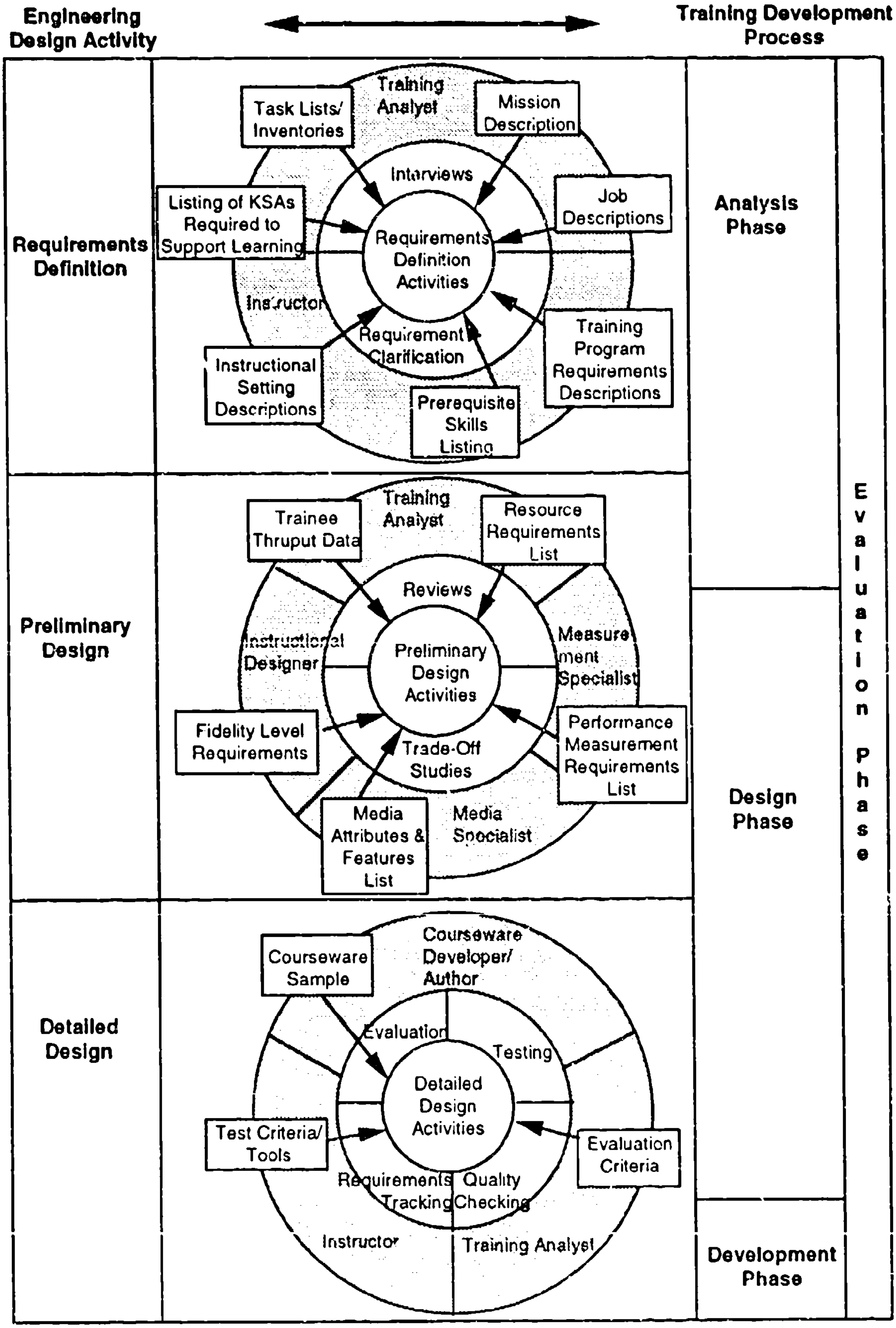


Figure 4 Management Structure

REFERENCES

1. Sorenson, B. H., and Park, J. S. (1990). *U.S. Joint Service Systems Approach to Training Design.* Brooks Air Force Base, TX: Air Force Systems Command.

2. Department of the Army. (1988). *Systems Approach to Training* (TRADOC Reg 350-7). Fort Monroe, VA : U.S. Army Training and Doctrine Command.

3. Department of Defense. (1990). *Military Standard Military Training Programs* (MIL-STD 1379D). Washington, DC.

4. Blanchard, B. S., and Fabrycky, W. J. (1981) *Systems Engineering and Analysis.* Englewood Cliffs, NJ: Prentice-Hall, Inc.

5. Wexley, K.N., and Latham, G.P. (1981). *Developing and Training Human Resources in Organizations.* Glenview, IL: Scott, Foresman and Company.

6. Abella, K.T. (1986). *Building Successful Training Programs: A Step-By-Step Guide.* Reading, MA: Addison-Wesley Publishing Company, Inc.

ADDITIONAL REFERENCES

Chase, W. P. (1974). *Management of Systems Engineering.* New York, NY: John Wiley and Sons, Inc.

Hatley, D. J., and Pirbhai, I. A. (1987) *Strategies for Real-Time System Specification.* New York, NY: Dorset House Publishing Co., Inc.

Department of the Air Force. (1986). *Instructional System Development* (AFM 50-2). Washington, DC.

Department of the Army. (1979). *Job and Task Analysis Handbook* (TRADOC PAM 351-4[T]). Fort Monroe, VA : U.S. Army Training and Doctrine Command.

DeVries, P. B. Jr., Eschenbrenner, A. J. Jr., and Ruck, H. W. (1980). *Task Analysis Handbook.* Brooks Air Force Base, TX: Air Force Systems Command.

Gagnr R. M., and Briggs, L. J. (1974). *Principles of Instructional Design.* New York, NY: Holt, Rinehart, and Winston.

Hays, R. T., and Singer, M. J. (1989). *Simulation Fidelity in Training System Design: Bridging the Gap Between Reality and Training.* New York, NY: Springer-Verlag.

Goldstein, Irwin L. (1986). *Training in Organizations: Needs Assessment, Development, and Evaluation.* Pacific Grove, CA: Brooks/Cole Publishing Company.

APPLYING THE INSTRUCTIONAL SYSTEM DEVELOPMENT (ISD) PROCESS IN U.S. AIR FORCE DEFENSE SYSTEM ACQUISITION

Ben H. Catalina
Southwest Research Institute
San Antonio, Texas

Conrad G. Bills
Loral Defense Systems
Akron, Ohio

ABSTRACT

The United States Air Force (USAF) has completed a new series of guides for designers of instructional systems—Air Force Handbook 36-2235, Information for Designers of Instructional Systems. Volume 3, Application to Acquisition, covers the major phases of the instructional system development (ISD) process and addresses them to the various phases of defense system acquisition. The ISD process has application in all acquisition phases, but the major effort occurs between the demonstration and validation phase and the production and deployment phase. The new Air Force ISD model incorporates the necessary functions for fielding successful total training systems. Fielding a new defense system with a total training system is a project that requires considerable management, coordination, and integration. Interface of the ISD process with the system engineering process ensures that critical functions are not overlooked early in the overall design and that these requirements are tracked throughout the acquisition for full implementation and life cycle support. This guide incorporates lessons learned from the past, applying a systematic, orderly process of integrated product development and treating ISD and system acquisition as a total system.

This paper discusses this new application of the ISD process in acquisition, the redefinition of activities leading to a common terminology for instructional designers and system engineers, and the orientation to quality improvement of the total training system throughout the life cycle of the defense system.

ABOUT THE AUTHORS

Ben H. Catalina is a senior project manager and instructional developer in the Instructional Systems Section. He has over 24 years of experience in training development, training management, and defense systems acquisition for the U.S. Air Force and industry. He has worked extensively with the USAF on the ISD project and is now serving as project manager for the field evaluation, revisions and new pamphlets development contract. He has a B.S. in Business Education from Mississippi State University and an M.Ed in School Administration/Supervision of Instruction from the University of Southern Mississippi.

Conrad G. Bills is senior training systems analyst for Simulation and Training Engineering, Loral Defense Systems. He has 20 years of experience with the U.S. Air Force operations and support programs in training and education, instructional system development (ISD), evaluation and administration, scientific management, and applied psychology. During the past seven years with the Air Force, he was senior training systems analyst for the Training Systems Product Group and Support Systems Engineering at the Aeronautical Systems Center, Wright-Patterson AFB, Ohio. He directed the contract for updating the Air Force ISD process, including the first volume for application of ISD in training system acquisition. He has an M.S. and B.S. in counseling and psychology/mathematics education from Brigham Young University.

APPLYING THE INSTRUCTIONAL SYSTEM DEVELOPMENT (ISD) PROCESS IN U.S. AIR FORCE DEFENSE SYSTEM ACQUISITION

Ben H. Catalina
Southwest Research Institute
San Antonio, Texas

Conrad G. Bills
Loral Defense Systems
Akron, Ohio

INTRODUCTION

Background

The application of the Instructional System Development (ISD) process in Aircrew Training System (ATS) acquisition for the U.S. Air Force brought to attention the need to update the Air Force ISD process. A baseline analysis was accomplished across the operational and support communities which apply the ISD process. A recommendation to the Air Force based on the results of the baseline analysis was that separate guidelines be developed for major instructional applications to address the unique requirements of each, written in a style suitable to the users' needs. The unique performance requirement defined for acquisition was the interrelation of engineering and training information. As the result of this recommendation, the Field Test AFP 50-68, Volume 3, Information for Designers of Instructional Systems–Application to Acquisition, was developed. This volume summarized the charted results of the Aeronautical Systems Center (ASC) Courseware Process Action Team (PAT). The Courseware PAT charted the courseware development process occurring within the system engineering process for total training system acquisition. This process was compiled and then coordinated with the government/industry steering group for training system acquisition. (During the field test of AFP 50-68, the Air Force publications office changed the nomenclature to AFH 36-2235.)

Volume 3 is the user's guide designed for personnel to use while applying the ISD process in defense system acquisition. It is intended to be an easy-reading document designed for the ISD novice as well as the veteran. Its purpose is to incorporate many applicable regulations and manuals into a single document that covers the phases of the ISD process and addresses them to the various phases of defense system acquisition. Volume 3 treats ISD with acquisition as a **total system**, incorporating the principles of integrated product development (IPD). There is considerable emphasis on system integration tasks and tasks not typical of ISD. The key to all of these tasks and ISD is integration to the **total system**. This paper provides an overview of Volume 3, focusing on some of the atypical aspects of ISD and defense system acquisition.

Although the ISD process has application in all acquisition phases, the focus of Volume 3 is where the major effort occurs between the demonstration and validation phase (I) and the production and deployment phase (III). Since the acquisition of major defense systems can routinely take ten years or more, it is imperative that one learns how to apply the phases of ISD with the phases of acquisition. This is equally important for modifications to current systems. Frequent coordination and evaluation are a requirement of success, as is revisiting of prior efforts and modifications where required. Figure 1 depicts the acquisition life cycle milestones and phases.

Lessons Learned

There is a definite contrast between the early application of ISD in defense system acquisition and the process applied in Volume 3. In the early application of ISD, the consideration for training was often an afterthought and was treated as part of the logistics elements important after the system was fielded. The burden of integrating a training system was on the operational command. ISD organizations were set up to begin preparation of the training curriculum in a time frame that closely

corresponded to the fielding of the defense system. These organizations quickly recognized the importance of obtaining long-lead items such as training simulators well in advance of the first defense system delivery. For example, the F100 jet engine (used in the F-15 and F-16 fighter aircraft) was not available for training purposes until eight years after deployment. As a result, efficient and effective maintenance was not available. In another example, operational E-3A AWACS (Airborne Warning and Control System) aircraft had to be used as trainers because trainers were not purchased with the defense system. In contrast, the application of the process in Volume 3 in a total quality environment resulted in the reduction of courseware development time by 40 percent for the C-17 ATS. The first crews trained in the C-17 training system were received by the test force at Edwards AFB, California, and were complimented by the test force as being the best-prepared crews ever. Delivery of the integrated training system, including full-mission simulation, was at the home base when required. The application does what the process was designed to do in the first place—improve training effectiveness and efficiency.

TOTAL TRAINING SYSTEM

A total training system is all-inclusive for meeting the training requirements. The training system is systematically developed to include the entire life cycle curriculum as well as the courseware, classroom aids, training simulators and devices, and operational equipment to present the curriculum. The training system also includes the personnel and logistic support to operate, maintain, or employ a defense system.

Fielding a new defense system with a total instructional system is a project that requires considerable management, coordination, and

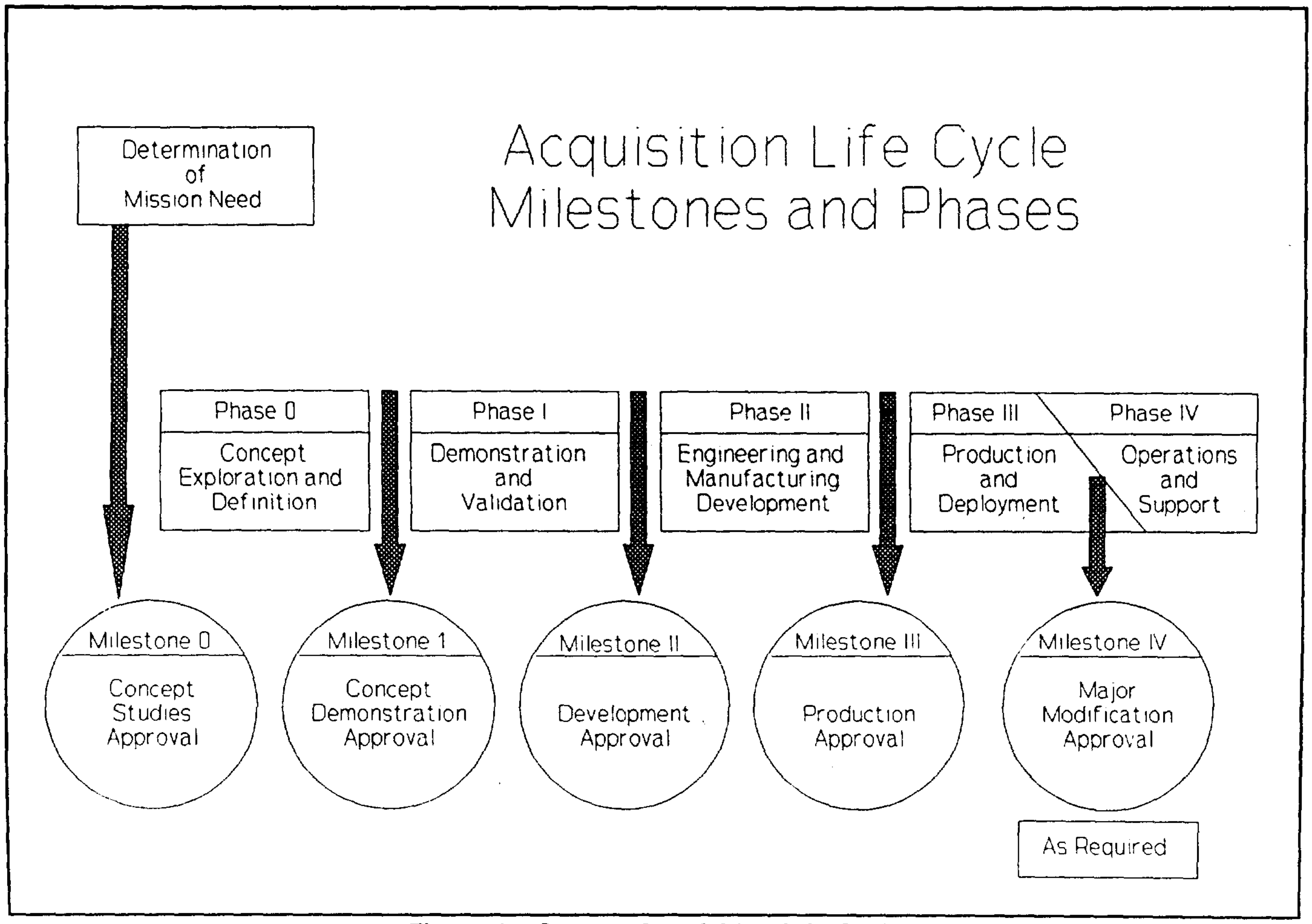


Figure 1. System Acquisition Life Cycle

integration. Lessons learned in fielding total instructional systems have shown that organizations responsible for integration of the system have often been left scrambling. Why? Because important and sometimes even critical functions were overlooked early in the overall design. The shortfalls range from "common sense" such as failing to analyze student production requirements, to "technical" such as improper integration of out-the-cockpit visual system design with the design of the simulator. Analysis of successful programs concluded that there are basic top-level functions required for operation of a total instructional system.

System Functions

The basic top-level functions must be in place before a training system can operate. These system functions, shown in Figure 2, include **management, support, administration, and delivery, and evaluation** which occurs throughout the process.

Management is the function of directing or controlling all aspects of the instructional system. These activities are an integral part of conducting instruction. **Support** includes those activities that provide for and maintain the system on a day-to-day and long-term basis. This includes long-range planning as well as day-to-day activities. **Administration** is the part of management that performs the day-to-day tasks of operating an instructional system. This includes functions such as documentation, student assignments, and student records.

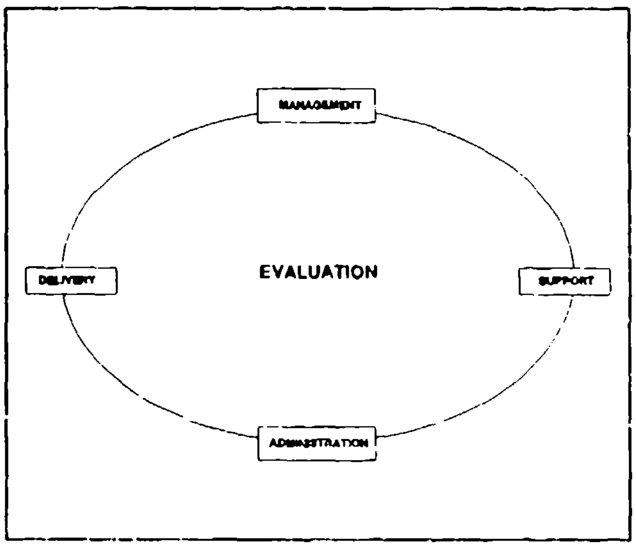


Figure 2. System Functions

Delivery is the means of giving students the instruction. Instructors, computers, and textbooks are examples of ways to deliver instruction. **Evaluation** is the continuous process of gathering feedback data through formative, summative, and operational evaluation to assess the system and, most importantly, assess student performance.

Using these essential functions to design the overall training system architecture and then allocating them to the respective system components, or people responsible, ensures that these functions are operational when the total instructional system is fielded. ISD products are integrated into the total system, and aspects of the system functions are active throughout all phases of the ISD process.

ISD Phases

The ISD phases used in the systems approach are **analysis, design, development, and implementation.** Evaluation activities are integrated into each phase of this process. To summarize these phases:

- **Analyze** and determine what instruction is needed.
- **Design** instruction to meet the need.
- **Develop** instructional materials to support system requirements.
- **Implement** the instructional system.

It must be emphasized that evaluation is a central function that takes place at every phase. ISD is a continuous, systematic process with continuous evaluation. ISD in the Air Force is used as a tool to ensure that quality systems are built to the customer's satisfaction. It helps managers and instructional developers build programs that teach what Air Force people need to know, when they need to know it, in the most effective and most efficient manner possible.

Quality Improvement Process

The ISD process implements all of the principles of the Quality Air Force (QAF) program. Quality is the vehicle to ensure that instructional systems are built and delivered customer centered. Quality improvement (QI) is the continuous, organized creation of beneficial change. It occurs throughout the ISD process. The updated ISD model, shown in Figure 3, depicts the interaction

of the ISD phases with the system functions and quality improvement.

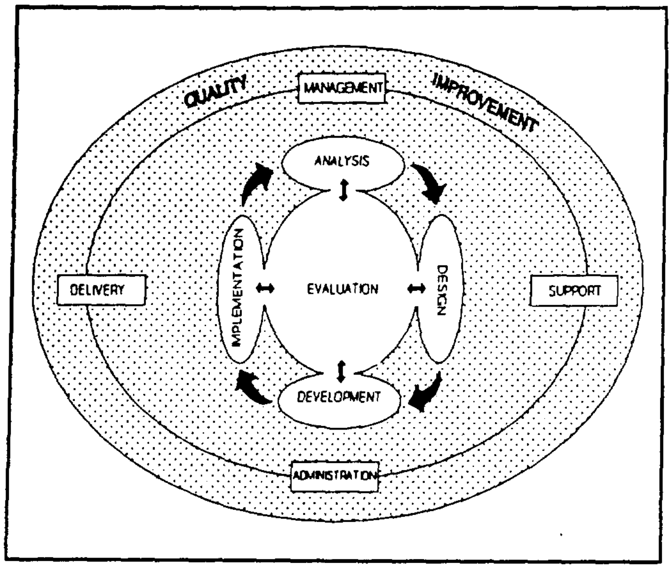


Figure 3. Updated ISD Model

APPLICATION TO DEFENSE SYSTEMS ACQUISITION

Manpower, personnel, and training (MPT) issues cycle throughout the entire weapon system acquisition process. The result of effectively handling these MPT issues can be concurrent delivery of these support elements with the delivery of the defense system. Once delivered, these MPT elements are sustained with the defense system throughout its life cycle.

ISD is the process for managing the acquisition of the training system for the defense system. This training system must be developed in the context of manpower and personnel estimates as well as defense system hardware and software design. Since the training system must be current with the weapon system design, development and production, and then systems engineering must also address the interface of ISD.

System engineering is a process which has been used for systematic development of the defense system as well as the training device hardware and software. The recent expansion of the training system concept to encompass the full life cycle of training for aircrew and maintenance personnel within a defense system acquisition has brought about an integration of the traditional ISD process within the system engineering process. This makes training system development a part of the integrated product development (IPD) team. Preliminary to the formation of the IPD team, the operational command forms a training planning team (TPT).

Training Planning Team

The Air Force recognizes the need for coordination and integration and requires that a Training Planning Team (TPT) be formed early in the acquisition cycle. A TPT is defined as an action group composed of representatives from all pertinent functional areas, disciplines, and interests involved in the life cycle of a specific defense training system. For a new acquisition, the TPT is formed at pre-concept and continues throughout the acquisition and day-to-day operation of the training system. The personnel on the TPT represent the using command, the system program office, and other concerned agencies. The TPT develops and uses the System Training Plan (STP) to ensure that training considerations, constraints and opportunities are adequately addressed in the defense system acquisition modification process.

The primary objective of the training planning team is to get the right agencies communicating and coordinating from the very beginning as a team. Once a System Program Office (SPO) is formed, the TPT bridges between the SPO and the operating command. The goal is to develop the STP and keep it current throughout the life cycle of the defense system.

Likewise, the primary operating command will establish and chair TPTs throughout the life cycle of the defense system. While the TPT may not meet every day, every week, or even every quarter, they will meet frequently enough to evaluate changes in the defense system for their effect on the training system. The TPT will update the STP annually or when changes occur that affect training in:

- Tactics
- Personnel
  - Structure
  - Demographics
  - Manning levels
- Defense system
  - Hardware
  - Software
  - Subsystem

- Training assets availability
- Funding priorities/levels
- Basing
- Operating commands

The TPT develops and implements alternate training strategies until the training system becomes current again with the defense system.

Whenever possible, advance notice of changes should be provided to the TPT to allow training of personnel prior to implementation of defense system changes.

System Engineering Interaction

With a properly operating Training Planning Team and a System Training Plan that is kept current, proper interfaces should be occurring with other defense system acquisition and life cycle support functions continuously. One important way that the ISD process meshes with the defense system is through interaction with system engineering. An "interaction" is a two-way street: ISD and system engineering communicate and support each other. But why is it important and how does it happen? First of all, a system is a composite of skilled people and equipment (hardware and software) that provide an operational capability to perform a stated mission. As mentioned earlier, ISD is the systematic process employed to design and develop training for a defense system.

The system engineering process is a logical sequence of activities and decisions transforming an operational need into a description of system performance parameters and a preferred system configuration. System engineering must consider personnel, the skills they require, and the training program to teach these skills as integral parts of the defense system. Failure to integrate ISD into system engineering can result in an inadequately supported system.

System engineering addresses those training system design issues having to do with translation of training system functional requirements (stated by ISD) into hardware and software. It considers the defense system hardware, software, support equipment, operations, and maintenance concept. System engineering examines new technology, similar systems, and existing systems to arrive at a functional description of the system in terms of hardware and software requirements. The system engineering process is used to produce the management and design decisions and data upon which the training system is based. ISD alone cannot fulfill all the needs of a total training system.

ISD and system engineering are two complementary processes that are used to design and develop training systems for defense systems. The processes have many similarities and each process accomplishes functions not accomplished by the other. All individuals involved with acquisition must ensure that ISD is considered in system engineering and vice versa. Many avenues exist for this interaction. Among them are:

- Acquisition strategy
- Training planning teams
- System training plans
- Integrated Manpower, Personnel and Comprehensive Training & Safety (IMPACTS)
- Requests for Proposal (RFP)
- Logistic support plans
- Logistic support analysis
- Technical interchange meetings
- Quality control
- Test plans
- Design reviews
- Program development plans

System engineering reviews of system requirements (SRR), system design reviews (SDR), preliminary design reviews (PDR), critical design reviews (CDR), and others should include instructional system reviews. Functional configuration audits (FCA) and physical configuration audits have a corresponding courseware readiness review (CRR). Tradeoffs are necessities in system engineering. This includes instructional system options considered at each phase of the process. Design decisions are reflected not only in hardware and software but also in courseware.

Acquisition Strategy

At a point when the TPT is formed and the STP is being written, a preliminary decision will be made on whether or not to contract for all or parts of the training. Assuming the decision is to have contractors develop at least a part of the training, the command with program management responsibility will develop an acquisition strategy. The acquisition strategy is finalized before each contracted activity.

In developing an acquisition strategy, the following should be considered by the SPO in coordination with the user.

- Current federal acquisition regulations
- Funding availability and constraints
- Defense system schedules
- Complexity of training system
- Types of training being acquired (operator/maintenance/other)
- Sole vs. multiple sourcing
- Lease vs. purchase
- Trained personnel requirements
  - How many?
  - When needed?
- One-time course vs. life cycle use
- Total contractor training vs. turnkey (using command operation)
- Other considerations

Getting the "big picture" is important in developing the acquisition strategy. The total instructional system perspective is needed to understand its full scope and how the integration will take place in order to have a fully operational system. Though a contracted activity may be treated as independent, the tie into the "big picture" ensures a good fit. Always consider how the instructional system fits into the overall defense system acquisition. Choosing the wrong acquisition strategy not only affects the instructional system, but can also cause delays in the defense system testing, support, and initial operational capability.

Evaluation

Evaluation occurs throughout the ISD process. Once instruction has been conducted, the Air Force will be specifically concerned with determining how well the training is achieving its objectives. Evaluation is the feedback that helps ensure that training objectives are achieved and the quality of graduates' performance is acceptable. The process continuously evaluates the course to determine if it is operating as designed. For example, six months after students graduate, are they still able to meet job performance requirements? If not, why not? Is it because of shortfalls in the course? Have mission requirements changed? Should changes in the course be undertaken? These are the kinds of questions you must ask and reviews you must make to ensure that the training that was developed is effective and efficient.

While evaluation occurs throughout the ISD process, **formative evaluation** should start early and continue through development, production, and test activities. It is the period from the beginning of planning to course readiness review or validation of materials.

One purpose of the formative evaluation period is to evaluate lesson/course development during the "formative" stages. It allows for corrections (remedies) to be made before training is fully implemented. It also includes acceptance testing of equipment and software, performance verification of system components, and assessment of the overall training system integration.

Summative evaluation begins at the Courseware Readiness Review, overlaps the formative evaluation period, and terminates at the Training System Readiness Review.

During summative evaluation, the training system is tested in the operational environment to validate the requirement baseline and assess the "summed" effect of the total training process.

During summative evaluation, questions are answered, such as:

- How well has the training been accomplished as reflected by operational requirements?
- Do graduates of a course meet established training system and operational performance standards?
- Are the training system performance standards correct?
- How can the training be better accomplished?

The primary purpose of summative evaluation is to determine whether the training developed for the students is effective and efficient. It is the process of collecting data from students, instructors, and other key evaluation interfaces as they use instructional media in the actual training environment. Its purpose is also to identify instructional materials, training media or instructional management system components that result in poor learning, inefficiency, or poor student acceptance. This data will then drive improvements.

Internal and external evaluation are categories of evaluation. Internal is within the training system and external is outside the training system. Inter-

nal and external evaluation activities occur within summative and operational evaluation.

The key difference between summative and operational evaluation is a matter of degree. The evaluation activities in summative evaluation are very intense and look at every possible data input. A review is conducted daily (if not more frequently) to assess the "bugs" still in the system and get them worked out as quickly as possible. Once the training system begins to stabilize, then the more routine period of operational evaluation begins. Data is collected selectively in order to keep a pulse on the entire system and its individual components. When areas of attention are raised in importance, then more intense data collection is accomplished only on that area and then phased back as the need is met. Day-to-day evaluation is a reflection of the evaluation activities during summative evaluation but is less intense. Evaluation continues both internally and externally using some of the same methods developed for summative evaluation.

Internal evaluations can be conducted by reviewing:

- Course documents
- Resources
- Instructional facilities
- Instructor performance
- Measurement programs
- Other sources as necessary

External evaluations can be conducted by:

- Questionnaires
  - For graduates
  - For supervisors
- Field notes
- Job performance evaluations

Operational evaluation begins at the conclusion of summative evaluation and continues throughout the life of the fully operational training system. Evaluation occurs on a system regardless of whether it is contractor- or USAF-operated. Evaluation in this period is similar to summative evaluation except it is less intense and reflects long-term operational data.

The purpose of operational evaluation is to provide real-time data for use in reviews, updates and quality improvement of training systems. It is continuous improvement.

The training source and the contract determine who conducts the operational evaluation.

Operational evaluation is a general reflection of the detailed procedures and data collection begun in summative evaluation but is more selective in data which continues to be collected. It usually starts at the Training System Readiness Review and lasts throughout the life cycle of the program. The emphasis shifts from establishing the instructional value of the courses to detecting flaws or deterioration. The primary goal is to maintain and improve course quality throughout the program life cycle. The following issues should be addressed in operational evaluation:

- Measurement and assessment of student learning in comparison to established training requirements and objectives
- Measurement of terminal objectives (qualification/certification)
- Identification and resolution of discrepancies and deficiencies in courseware and the training system
- Assessment of training in light of modification/upgrades in the defense system

Operational evaluation continues by both internal and external means, using to some degree the same methods developed in summative evaluation.

SUMMARY

The ISD process is really a derivative of the system engineering process and can be integrated in the acquisition of training systems for new defense systems. The once-perceived disconnect between the ISD community and the system engineering community is more semantic than real, and AFH 36-2235, Volume 3 provides the common understanding for both communities. This application should be a part of the integrated product development process and should be managed within the training system product group for the life cycle of the defense system.

The process of total training system design begins with the basic training system functions. The training system functions are key to building the overall training system architecture, assuring that all these functions necessary for successful

operation are in fact fielded for implementation. The tracking of the training system design through the system engineering reviews ensures that the system and component level specifications map the functions into requirements. The test and evaluation of the training system and its components assures that integration of the training system functions occurs, requirements are met, and the total training system becomes operational as designed.

The quality mindset throughout the acquisition and on throughout the life cycle of the defense system keeps the dynamic processes of ISD and system engineering active. Appropriate phases of these processes are entered as the feedback from the internal and external evaluation activities gives indications for a need to improve. Continuous improvement results in a training system which meets the needs of the user and continues to be effective as well as efficient.

BIBLIOGRAPHY

Air Force Manual 36-2234, Instructional System Development.

Air Force Handbook 36-2235, Information for Designers of Instructional Systems, Volume 3, Application to Acquisition.

Bills, C.G. and Butterbrodt, V.L. "Total Training Systems Design to Function: A Total Quality Management Application." Proceedings of National Aerospace Electronics Conference (NAECON), Dayton, OH, 1992, 1018-1023.

Golas, K.C., Shriver, S., Bills, C.G., and Bowden, P. "Update of the U.S. Air Force Instructional Systems Development (ISD) Process." Proceedings of the Interservice/Industry Training Systems and Education Conference (I/ITSEC). San Antonio, TX, 1992, 652-662.

Reconsidering the Role of ISD

Kathy D. Andrews, Ph.D.
Information Spectrum, Inc.
Falls Church, Virginia

James M. Young, Ed.D. & Larry W. Barks
Naval Air Systems Command
Washington, D.C.

ABSTRACT

Two decades of military experience with ISD have yielded mixed results. Depending on one's perspective, "doing ISD" may still be considered essential to the development of effective, efficient training systems or it may be regarded as a resource-consuming chore to be avoided to the extent possible. Both perspectives and numerous variations have merit. This paper examines some of the problems associated with ISD models and their applications and discusses potential solutions, including redefining ISD's role. The problems with ISD, the acquisition process, and Navy training in general are not simple, and filling the knowledge gaps, streamlining processes, and producing better-equipped ISD practitioners are only partial solutions. Although the paper focuses on naval aviation, it is applicable to other naval activities and military services.

ABOUT THE AUTHORS

Dr. Andrews is the Vice President, Training Systems Group at Information Spectrum, Inc. In addition to quality improvement responsibilities, she serves as technical advisor on several training system acquisition and development projects. Prior to joining Information Spectrum, Inc., Dr. Andrews served as an Aerospace Experimental Psychologist in the Navy. She holds a Ph.D. in experimental psychology from the University of Georgia.

Dr. Young is assigned to the Naval Air Systems Command, PMA205, as the Program Manager for the U.S. Navy's "A" school basic skills program, the Aviation Physiological and Water Survival Training Devices program, and General Training programs which include aircraft launch and recovery training devices like the Landing Signal Officer Training program, the Electronic Warfare Training devices program, and the Air Traffic Control basic and proficiency training programs. Additionally, Dr. Young manages the active and pending Foreign Military Sales (FMS) training programs. In directing the joint services system acquisition program, Dr. Young is responsible for ensuring cooperation in identifying common functional requirements, including military capabilities and operating features, to foster development of policies and procedures which will lead to joint service procurement of interchangeable components and sub-systems. Dr. Young received his Doctorate of Education from East Texas State University in Commerce, Texas.

Mr. Barks is assigned to the Naval Air Systems Command as Manpower, Personnel and Training Programs Manager. He has over 16 years of training acquisition management experience with that command. He manages the execution of the HARDMAN program for naval aviation, development of Navy Training Plans, and the Instructional Systems Development (ISD) application for emerging Naval aviation weapons system. He also serves as the Naval Air System Command representative for development of Military Standards related to training.

Reconsidering the Role of ISD

Kathy D. Andrews, Ph.D.
Information Spectrum, Inc.
Falls Church, Virginia

James M. Young, Ed.D. & Larry W. Barks
Naval Air Systems Command
Washington, D.C.

INTRODUCTION

The acquisition and fielding of a large-scale training system in parallel with a weapon system can involve years of effort, thousands of tasks, hundreds of people, and a paper trail that extends for miles. Managers and performing agencies juggle the demands of budget, schedule, and reporting requirements while undoing and redoing training system efforts in response to changes in the parent weapon system. High-cost items, regardless of the role they will ultimately play within the training system, typically consume the major share of resources and attention. Weeks or months can be spent negotiating compromises and changes. Resolution of one problem may create others, and the cycle of compromising, undoing and redoing starts again. It doesn't take long for an effort to derail, and it's easy to lose sight of the overall objectives.

Still, training systems get developed and fielded--not necessarily within budget, on schedule, or in a form that remotely resembles the original design, but fielded nonetheless. And, over time, most military people learn to perform their jobs reasonably well. How this happens may have little to do with Instructional Systems Development (ISD). To some, "doing ISD" is considered essential to the development of effective, efficient training systems. To others, it is a resource-consuming chore to be avoided to the extent possible. Both perspectives and numerous variations have merit. This paper examines some of the problems associated with ISD and makes a case for redefining its role in the acquisition of Navy training systems. Simulators and other training devices will still be purchased, academic courses will still be developed, and people will still be trained with or without the time and expense of ISD. "Why do ISD?" turns out to be an interesting question. This paper focuses on naval aviation, but it is also applicable to other naval activities and military services.

THE MILITARY TRAINING ENVIRONMENT

Military training systems are rarely static, and their modification usually continues throughout the life cycle of the parent system. Within naval aviation, as a result of aircraft Engineering Change Proposals (ECPs), new tactics, or the identification of training deficiencies, new requirements evolve and training system elements may be introduced or modified. Especially within fleet aviation, training programs must be adaptable--both to the projected needs of the next deployment and to the constant ebb and flow of training resources. When a simulator is down for modification, bombs aren't available, or range time is limited, workarounds are implemented. Today's base closures, force draw-downs and realignments also require training workarounds and training system modifications. The required flexibility is an inherent part of the military environment.

Training programs must also be adaptable to changes in preceding or follow-on courses of instruction. For example, with the (potential) introduction of the T-45 to replace the T-2 and TA-4 in the undergraduate advanced jet training program, adjustments may be required both in the T-34 syllabus and in follow-on Fleet

Readiness Squadron (FRS) syllabi. Similarly, the introduction of the Joint Primary Aircraft Training System (JPATS) or other system to replace the T-34 may impact the T-45 syllabus and others.

It should be obvious to anyone who works on maintaining the training continuum within one or more pipelines that the changes made in one course of instruction to accommodate changes in another course are not necessarily made to improve or maintain training effectiveness. Being able to afford one change may mean making other changes for cost-savings purposes. The potential impact on training effectiveness usually isn't ignored--some kind of analysis is done to demonstrate that the change probably won't hurt or may even enhance training.

It was to this environment, in part, that ISD was introduced approximately 20 years ago. Neither the concept of a systems approach to training nor the procedures that became part of the ISD model were new. However, the first application within naval aviation of the "new" ISD model can be dated to 1974 and to some of the aircraft that are still being flown, e.g., the EA-6B, E-2C, and A-6E.[1] Then--as now--reactions to ISD varied. The words of McClelland, writing in 1978 about the previous 25 years' experience with the systems approach to training, still apply: "Today.....the full potential of applying the systems approach to training has not been realized."[2]

It is worth remembering that the basic objective of any training system is to enable people to develop the capabilities required to proficiently and reliably perform their jobs. The precise components that make up a system can vary considerably--even to meet the same training objectives--and people will still learn. An effective system is simply one that works. Even very inefficient training programs can help people develop the skills required to perform their missions--but at a higher cost and over a longer period than necessary. Clearly, in the resource-constrained military environment, training system development efforts must maintain a focus on both effectiveness and efficiency.

THE ROLE OF ISD: IN THEORY

On paper, the role of ISD in the development of a new training system is straightforward: ISD provides the logical framework and procedures for systematically identifying training system requirements and then translating these requirements into actual instructional materials, devices, courses, etc.

ISD consists of a series of interrelated activities, each of which is intended to provide part of the data required to produce an effective training program. This series of activities is generally divided into five phases: analysis, design, development, implementation, and evaluation (or quality control). The analysis phase entails the determination of tasks that must be performed to operate or maintain the parent system, entry-level skills of the future system operators and maintainers, and, based on those two sources of information, training requirements. During the design phase, the various training system elements (courses, trainers, etc.) will be planned. The development phase entails the actual production and tryout of training materials. The implementation phase involves putting the new or modified system in place in the field. The final phase, evaluation, is intended to ensure that the system continues to function as required throughout its use.

Also on paper, ISD is an iterative process that provides for the systematic refinement of training requirements and materials as more and more information becomes available. During the development of a new weapon system, changes in engineering specifications or, initially, limitations in available data may impact the identification of training requirements. More than one report (or other product) may have to be updated simply as the result of changing a single operator or maintenance task. This is not unusual, and provisions are generally made for the revision of ISD reports as required to reflect these changes. It is rarely the case that each report is done once and only once without updates. Automated systems simplify the tasks of managing developmental data, reports and courseware.

ISD also plays a role in the modification of existing systems. Typically on a smaller scale, the same sequence of activities that results in identification of training needs and resource requirements for a new system is repeated for an existing system. The cycle of analysis, modification as required, and evaluation will

continue (in some form or another) throughout the life cycle of the system.

Since the early 1970s, various ISD procedural models have been developed, and numerous handbooks, standards, and specifications have been published. Both within and across military services, different models are employed. For example, the Navy has typically used one approach for the development of aircrew training programs and another for aviation maintenance training. Despite methodological differences, the objective is the same: systematically identify and meet training requirements.

To ISD practitioners, the question "Why do ISD?" may at first seem incredible. But even the enthusiasts (most of them anyway) will agree that ISD in theory sometimes bears little resemblance to ISD in practice.

ISD IN PRACTICE: THE PROBLEMS

ISD is labor-intensive, time-consuming, and costly, but so is the process of acquiring a training system without the use of ISD, so these factors are hardly critical determinants of its worth. More importantly, the application of ISD to a training development effort provides no guarantee that an effective, efficient training program will result. Numerous factors may combine to limit both ISD's role and impact. Some of these factors are discussed below.

Lack of Expertise

For the most part, application of ISD requires the use of both training development specialists and specialists in the system being developed. The training development specialists, knowledgeable about training technologies, instructional strategies, and human behavior, set the pace for ISD efforts. The term subject matter experts (SMEs) applies to the aircrew and maintenance personnel who fulfill the role of system specialists. Some SMEs will be former military personnel hired by a contractor involved in the ISD effort, and others will be active-duty military who may be available at various points in the process to assist in the development effort. For example, the formation of an Instructional Systems Advisory Team (ISAT), which entails on-site availability of military personnel, provides regular opportunities for informal (and formal) fleet involvement in the ISD process. Members of the Fleet Project Team (FPT) can also play an essential role not only in the development of trainers but in the development of curricula.

SMEs are not required to be experts in ISD to perform their jobs well. They bring other qualifications (and problems) to a design effort and do not share responsibility for lack of expertise in ISD. It's the inexperienced, unskilled other specialists on the ISD team--the training system designers and developers--who pose major problems for ISD efforts. ISD is not an objective approach to training system development, and although ISD cookbooks exist, none would permit the inexperienced developer to (intentionally) construct a good program.

Decisions at each step in the ISD process entail combining knowledge of learning and instructional techniques with best judgment and best guesses. In the hands of inexperienced designers, the resulting training programs may be inefficient and of limited effectiveness. Unfortunately, technically competent, skilled, experienced developers sometimes seem to be in short supply. Unless they are also familiar with the military environment, their recommendations, sound as they may be, can be met by incredulous stares or simply ignored.

Insufficient/Excessive Procedural Guidance

MIL-STD-1379D (*Military Training Programs*) and many other ISD documents provide guidance for the experienced instructional systems developer on what should be done, but they do not describe the how. And the "how" can usually be approached in a variety of ways--none necessarily entirely satisfactory. For example, the specification of tasks to be trained seems like a simple enough undertaking. In practice, there are a variety of ways to construct task listings and numerous theoretical arguments over which is preferred and why. The experienced developer (if allowed) simply applies what will seemingly work best in the current project. Others must resort to whatever guidance they can find. Attempts to provide this guidance have often resulted in overproceduralization of ISD steps, with the same unsatisfactory results.

Form Over Function

In the absence of sufficient procedural guidance, performing agencies and/or government representatives may resort to microscopic

examination of what is available to glean enough information to make decisions. A carelessly constructed sentence in the Statement of Work (SOW) or Data Item Description (DID) or a sentence that usually but not always applies to an ISD effort is taken literally by one or both parties. The result is overemphasis on the form of a deliverable and insufficient attention to the purpose. To continue the example of the task listing, regardless of the structure used, not all human endeavors will fit neatly into the categories. So force-fitting is employed to meet the SOW or DID requirement, and the intent of various task statements or objectives may be distorted in the process. Since most ISD activities are interrelated, what is done at one point will impact what is done at the next, and inaccuracies or inconsistencies may be compounded as time goes on.

Force-fitting may also be employed unintentionally. Cookbooks, developed to fill the gap caused by insufficient procedural guidance and a short supply of trained developers, may provide step-by-step procedures for completing various tasks. These steps are themselves simplifications, and reliance on them again results in overemphasis on form.

Incomplete Applications

Inadequate and incomplete ISD efforts stem from several factors including inexperienced personnel. Even with strong team members, available resources are almost always inadequate to permit unconstrained design, development, and evaluations of training systems. As a result, the concept of a training "system" may still not be fully realized.

In addition, training system managers differ in their understanding of or willingness to apply ISD. To some, doing ISD means designing and developing the paper-based instructional material to be used to support classroom instruction. They are perfectly content to leave that to the "experts" while they manage the more critical elements of the system - facilities, training devices, and other hardware. Even believers in the ISD concept may be reluctant to apply the results of analyses if the results suggest some "new" approach to training or a "new" piece of training equipment.

Poor Timing

Many ISD efforts have failed to achieve the desired results because they have not been initiated at the proper point in the acquisition cycle nor synchronized with "real world" requirements. For example, trainer specifications may be developed well before the ISD effort is off the ground. Although the contractor may still be responsible for completing the required ISD steps--including media specification and development of trainer functional characteristics --the exercise is usually pointless. Similarly, the training system manager must plan for facilities requirements and other high-cost or long lead-time items. The number of classrooms and number and type of computers for computer-based training may well be decided long before the media selection process indicates such a system should or should not be employed.

Lack of Responsiveness

As suggested in the previous section, lack of responsiveness of ISD to the training system manager's requirements is in part related to ineffectual timing of the ISD effort. The manager must make decisions when acquisition milestones or other reporting requirements dictate, and the absence of ISD data doesn't change that.

Descriptions of the ISD process, for example in MIL-STD-1379D, typically do not explicitly relate ISD products to acquisition reporting requirements, and the instructional system developers may not know when decisions that will impact the shape and substance of the training program will be made. Even if they do know, someone on the team (not necessarily the instructional developer) may insist on rigid adherence to one or another set of ISD procedures, the timing will still be off, and the manager still won't get the data.

WHAT'S RIGHT WITH ISD?

It should be fairly obvious that the problems with ISD are only partly due to inherent deficiencies. ISD models are simply that - models or frameworks for instilling logic and order to a complex, sometimes very disorderly process. Applied knowledgeably, the ISD framework will always result in some improvement in the process. Whether the model is called ISD or SAT (systems approach to training) or anything

else, yes, some sort of systematic approach to identifying and meeting training requirements is essential to a sound acquisition program.

The development of effective, efficient training systems is art and science - just like flying, the practice of medicine, or diagnosing equipment failures. The experienced instructional designer, who brings to the team a large body of theoretical and applied research on learning, is an irreplaceable element in the design and development process. Despite gaps in the base of knowledge about human performance, enough evidence is available to predict for many tasks the effects of altering basic variables like time allotted to learn, amount and distribution of practice, and instructional methods and media. These variables can be intentionally manipulated to improve the fit between training needs and the demands that the military environment places on training programs, e.g., frequent turnovers of personnel; the requirement to schedule training in a way that personnel availability can be predicted; the need for a series of courses; the requirement to train large groups of people with widely varying backgrounds and skills; the need for flexibility, etc. Best guesses are still required, and the competent developer is not easily replaced.

TRAINING EFFECTIVENESS REVISITED

As indicated earlier, an effective training system is simply one that works, and many people might argue successfully that, in general, military training systems do work. Even a casual observer would have little difficulty distinguishing between the student pilot and the third-tour fleet aviator. The smooth, seemingly effortless performance of the proficient aviator bears little resemblance to the erratic performance of the novice who may be overextended initially even by the basic task of flight control.

Obviously people learn, but "how well", "compared to what", and "at what cost" are questions with no simple answers. The contribution of the formal training system to skill development--or of any single element of the system--is not easily measured. Although the Navy collects a variety of data that can be used to infer the effectiveness or efficiency of training systems, formal training system evaluations are rarely done. When they are done, the results are sometimes difficult to interpret or not used to impact future design efforts.

Even without precise measures of effectiveness or efficiency, it's not difficult to find evidence of an uneasy fit between training systems delivered to the field and the needs of the users. One doesn't have to look far for training materials that are shelved almost as soon as they are delivered, device design features or entire devices that are ignored, and unused or underutilized computer-based training systems. (Although there are a variety of reasons for these uneasy fits--including some good ones--many are not related to ISD and aren't the focus of this paper.)

Most people who have been through any kind of Navy or other DoD training pipeline need not look beyond their own experience to understand that people who are motivated to learn will do what they can to learn--in spite of the system. Students compensate for unprepared and ill-equipped instructors by conferring among themselves about the meaning of this concept or that, wading through texts, or asking other instructors. They also employ self-teaching of the trial and error variety on almost every piece of equipment in the inventory. Fortunately, more often than not, they and the equipment survive. But mishap rates and equipment repairs and replacements attest to the high costs of some of these experiments. Discussions in the ready room, around the coffee mess, chief to new sailor (or Ensign), sea stories, and so on also typically play a part in Navy training, sometimes compensating, in some sense, for the lack of transfer of information and skills through other channels.

Information about the successes and failures in training system acquisition may be only informally collected and not fed back into a new design effort. As a result, past mistakes tend to be repeated, and the basis time and again for some design decisions is simply: "It worked before (I think), so it must be okay to do it this way."

SOLVING PART OF THE PROBLEM

The problems with ISD (and the acquisition process in general and Navy training) are not

simple and won't be easily resolved. Although efforts to fill the gaps, streamline the process, and produce better-equipped practitioners continue, they'll take time. The implementation in 1990 of the new standard, MIL-STD-1379D, may have also introduced new problems, in part because of the lack of readily available guidance on its use. An additional problem stems from trying to integrate individual service/activity and program requirements into a standard. Current joint-service efforts to refine the standard and develop common data element descriptions will resolve some of the difficulties.

A partial remedy for inadequate ISD efforts is ensuring that qualified teams - both within and outside of the government - are available to work on the projects. And "qualified" means far more than having a string of degrees or "x" number of years of fleet experience. Another fix entails shifting our approach to the use of ISD.

RECONSIDERING THE ROLE OF ISD

It's perhaps worth repeating that the purpose of a training system is to help people develop the capabilities required to perform their missions. The ever-present resource constraints, the sometimes large pools and lengthy waits for additional training, insufficient knowledge or technology to always provide the best training solution, and the sometimes arbitrary decisions that negatively impact training systems all provide good obstacles to learning; it's hardly necessary to create more.

It is also worth remembering that all training system components are simply tools to facilitate the learning process or improve the efficiency with which training can be delivered. Training devices, books, instructors or other elements are the means for providing instruction and opportunities to practice developing skills. Some people argue that too much energy is dedicated to the hardware and software components of the system and not enough energy to the message. Although it's true that training device acquisition projects can take on a life of their own, the energy devoted to the cause is laudable.

Perhaps it's more appropriate to say that not enough attention also goes into integrating and shaping training devices and other components to ensure that the resulting training system will serve its intended purpose. The myriad of design decisions (and compromises) made during a development effort that impact the capabilities of a training device also impact the rest of the system. Where more or fewer capabilities than planned are the result of these decisions, more, fewer or different capabilities (not necessarily in a one-to-one relationship) must be considered for other system components.

Purposeful design requires maintaining a constant focus on what it is the system is intended to accomplish. This means identifying the training requirements beforehand (difficult as that sometimes may be) and applying what we know about learning and performance not only to the design of the curriculum but to hardware and software components as well. Center stage belongs mostly to training requirements, partly to issues of training efficiency, and only rarely to technology.

It is easy to lose sight of the overall objectives--especially if system design efforts follow device design efforts. Training objectives, a preliminary syllabus, the plan for other training tools, reasons for incorporating device design features, and concern for the ultimate users must precede, parallel, and shape the design and development of training devices.

Knowledgeable application of ISD--at the right time and in a way that meets the needs of the acquisition process--will require some adjustments to the way we do business. To effectively serve as it should, as a framework and road map for the system design effort, ISD must be responsive to the very real demands of budget, schedule, and reporting requirements and to the needs of various players for information in different forms. As an example, the revised Air Force ISD model represents a step towards addressing the different information requirements of different agencies by the inclusion of a series of guides for different user communities.[3] Training system managers do not need to know the difference between Miller's taxonomy of human tasks and Gagne's, but they do need to be able to ask the right questions of the ISD team. They, along with every other team member, also must be willing to suspend belief in what they know about training at least long enough to weigh the available evidence. The training development specialist must be

prepared to explain why one approach is preferred over another and to identify what constitutes best judgment, best guess, or empirically or theoretically sound design concepts. It's then up to the program manager to make the hard decisions.

Knowledgeable application of ISD also means recognizing the essential role played by potential users of the training system in the development effort. No, the users don't always know the best way to meet their own needs, but they provide insight into system and human considerations that others may overlook. Just as surely as motivated people will do what they can to learn, so, too, will they defy--to the extent that they can--all efforts to make them use what they don't want. User acceptance is a requirement in any design effort, and attempts to explain, demonstrate, and persuade should precede design decisions. When these attempts don't work, it's usually far better to reconsider the design approach than to expect that the impasse will eventually be resolved in favor of system or component use.

CONCLUSIONS

There are no technological solutions to the problems with ISD. Automated design and development tools, improved configuration management systems, and better integration of data from ISD analyses, logistic support analyses, and other sources will improve the products and streamline the process, but they will not ameliorate the inherent weaknesses in the ISD model. Nor are there substitutes within the discipline for the experienced ISD practitioner--who has an appreciation for both the art and science of systems analysis--or for the involvement of end users and subject matter experts.

The value of ISD stems in large part from its application as a framework and road map for the training system design effort. Without top level commitment to this concept and its execution, ISD may well be a resource-consuming chore.

Implications for Training System Managers

"Doing ISD" as if it were simply one other acquisition requirement--apart from facilities planning, device designing, and other major system design considerations--will ensure its limited utility. The systems approach to training encompasses all training system elements, including devices, and sound design decisions must stem from both an analysis (beforehand) of training requirements and an understanding of the interplay among the system elements being developed. Training system managers can help ensure the success of training system development and modification projects both by committing to the application of ISD as a road map (nothing less) and by insisting that the effort be responsive to the demands of budget, schedule, and reporting requirements. ISD is an iterative process--not a set of specific procedures that must be performed in only one way and completed in its own time. It is designed to assist the decision-maker, especially in instances of uncertainty, e.g., when available data from the system under development are limited. If the effort isn't doing this, then the team isn't doing ISD.

REFERENCES

1. Jasper, C.R. & Funaro, J.F. Application of ISD to emerging aviation weapons systems of the 80's. Proceedings from Conference on Application of ISD to Emerging Weapons Systems of the 80's. Society for Applied Learning Technology. October 1978.

2. McClelland, W.A. Current research programs in ISD. Proceedings from Conference on Application of ISD to Emerging Weapons Systems of the 80's. Society for Applied Learning Technology. October 1978.

3. Golas, K.C., Shriver, S., Bills, C.G., & Bowden, P. Update of the U.S. Air Force Instructional Systems Development (ISD) process. I/ITSEC Proceedings 1992.

EVOLUTION OF A TRAINING PROGRAM

The Effects of Simulation on the MH-53J PAVE LOW Combat Crew Qualification Course

Major George Selix
542d Crew Training Wing
Kirtland AFB, NM

ABSTRACT

This paper showcases the effectiveness of enhanced training through simulation by examining the evolution of the MH-53J Combat Crew qualification course over a 7 year period (1986 to 1993). The MH-53J PAVE LOW is the primary special operations helicopter asset in the US Air Force. The qualification course in 1986, was almost totally reliant on lecture and "on-aircraft" training. In three years, a high fidelity Weapon System Trainer/Mission Rehearsal System, an Operational Flight Trainer, and numerous part task and computer based training devices have been added to the training program. Integration of Advanced Training Devices into the syllabus has also allowed us to absorb a 20% cut in MH-53J flying hours. Our examination of the Pave Low course concentrates on the integration of simulation at different levels of training program and the decisions on simulator fidelity and procurement that made it all possible. Some lessons learned for other aircrew training programs are included at the end of the paper, dealing with sensor correlation, databases and training transfer issues.

ABOUT THE AUTHOR

Major George Selix is currently Chief, Rotary Wing Training Systems Division, 542 Operations Group, 542 Crew Training Wing, Kirtland AFB, NM. Responsibilities center on acquisition and integration of training devices into the MH-53J and MH-60G training programs. He is qualified as a night tactical Flight Examiner in the MH-60G. Previous assignments include the Combat Aircrew Training School at Nellis AFB, NV, Operational Test and Evaluation pilot at Edwards AFB, CA, and Chief of Tactics and Training at the 55 Special Operations Squadron, Eglin AFB, FLA. Major Selix holds a B.S. in Business from Norwich University and an M.S. in Industrial Management from Clarkson College.

EVOLUTION OF A TRAINING PROGRAM

The Effects of Simulation on the MH-53J PAVE LOW Combat Crew Qualification Course

Major George Selix
542d Crew Training Wing
542d Operations Group

INTRODUCTION

The MH 53J Pave Low helicopter is the primary rotary wing platform for the Air Force Special Operations Command (AFSOC). The mission of the Pave Low is infiltration and exfiltration of various 'customers' through denied territory in support of a theater commander's contingency or wartime taskings. To accomplish this mission, the Pave Low is equipped with a variety of mission sensors and defensive systems including Terrain Following/Terrain Avoidance Radar (TF/TA), Forward Looking Infrared (FLIR), projected map display (PMD), ALR-69 Radar Warning Receiver, ALE-40 flare system, and IR and Radar Jammers. It is the mission of the 542d Crew Training Wing (542d CTW) to provide a steady stream of combat ready pilots, flight engineers, and aerial gunners to three fielded units in support of the Pave Low mission.

The 542d Crew Training Wing, formerly the 1550th Combat Crew Training Wing, is charged with mission qualification training for all Air Force rescue and special operations rotary and fixed wing aircrew members.[1] Prior to 1986, initial H-53 aircraft qualification was done at Kirtland AFB, while mission training for MH-53H crewmembers was accomplished at the Central Training Flight (CTF) at Hurlburt Field, Fla. In 1989, after the conversion of the MH-53H to the MH-53J Pave Low, initial aircraft and mission qualification training was consolidated at Kirtland AFB. The curriculum was built by combining portions of the previous Hurlburt and Kirtland courses. The Pave Low aircrew training course is now, with 150 programmed training days, the longest combat crew qualification course currently being taught in the Air Force. The tasks, conditions, and standards against which the students are trained include correlated sensor (TF/TA and FLIR) operations at night, adverse weather, and in an active threat environment. The current curriculum is a product of iterative task analysis and the acquisition and subsequent integration of numerous low, medium, and high fidelity aircrew training devices. This study follows the evolution of this training program and provides some valuable observations for trainers, program managers, and users, especially those looking to integrate ATDs upgrade their current technological base.

1986 MH-53H Training Course.

The MH-53 training program was, in 1986, primarily an aircraft based training course. The only imbedded aircrew training device was the HH-53C trainer. This trainer had an inoperative 6 degree of freedom simulator with a Vital-III-6000 visual system and basic H-53 avionics but lacked mission avionics and sensors. Because of this, integration into the training program was generally confined to basic aircraft procedural training (start-up, shutdown, and emergency procedures) and instrument training. Procedural training was done on static ramp aircraft. Table 1 shows the break-out of instructional time by course objectives and training medium. Because of a high aircraft utilization rate required to support flying training, procedural training was, by necessity, conducted very early in the morning (0400 seat time). This had the expected negative impact on student performance.

---

[1] Currently, MC-130E and AC-130H training is conducted at Hurlburt Field, FLA. MC-130E training will move to the 542d CTW in FY94

Job Element	Academic	PT	Sim	Flight	Msn Support	Total
AC duties	1.0	-	-	-	-	1.0
Flt Prep	15.1	-	-	-	0.4	15.5
Pre-depart	1.7	7.0	4.0	-	4.0	16.7
VFR Proc	6.9	-	-	9.9	14.9	31.7
IFR Proc	10.4	-	14.0	1.5	10.5	36.4
Adverse Wx	1.2	-	-	-	0.6	1.8
Hoist	4.5	-	-	5.2	7.1	16.8
Sling	3.1	--	1.4	2.8	7.3	
Formation	1.6	-	-	2.3	3.4	7.3
Tactical	24.0	-	-	8.9	16.7	49.6
Remotes	7.0	-	-	5.9	8.2	21.1
FCF	1.2	-	0.5	-	-	1.7
Search	5.8	-	2.0	1.8	3.7	13.3
Air Refuel	2.1	-	1.5	6.0	10.4	20.0
NVGs	5.0	-	-	-	-	5.0
EPs	13.8	-	6.0	5.1	10.5	35.4
TOTAL	**104.4**	**7.0**	**28.0**	**48.0**	**93.2**	**280.6**

TABLE 1. 1986 H53P1 Course

IMPLEMENTATION DECISIONS

The MH-53J is an order of magnitude more sophisticated than the basic H-53. The qualification course faced increased training requirements at the same time that flying hours and ramp aircraft were to decrease. The obvious answer was to acquire and integrate a series of more capable training devices into the new MH-53J curriculum. To do that, a number of significant implementation decisions were made. Although I'd like to say they were marvelously insightful and sage, especially in the early stages, and that our perfect acquisition plan was followed without deviation throughout implementation, in actuality these decisions were made iteratively during the ramp-up of the training program and continue today. Mid-stream corrections, which were necessary to keep the training program aligned with user requirements, muddied the acquisition process by providing a moving baseline.[2] Decisions on conduct of training and curriculum were almost always directly responsible for decisions in acquisition. The reverse, however, was often true: changing technology drove changes to the curriculum. This was particularly evident when looking at the MH-53J Weapon System Trainer (WST). The following discussion represents a summary of the major decisions made in building the Pave Low training program.

1. Off-load as many training events as practical to simulation. This decision was influenced by three complimentary factors. First, aircraft flying time is and will continue to be expensive. So, from an annual operating budget perspective, high fidelity WST simulation at $800-$1000 per hour is much more cost effective than aircraft flying hours at $3100 per hour.[3] Simulators also prove to be more reliable, hence more available than aircraft. Studies show that mature simulators are better able to provide system and sub-system training compared to operational aircraft.[4] Simulators are also not subject to environmental conditions that often restrict flight training (icing, thunderstorms, etc.). This allowed us to 'tighten' up the course by removing alternate mission or weather days. Second was the effectiveness of simulator training time. Most of our simulator training sorties place both student

[2]USAF Helicopter Training High Fidelity Simulation, Lt Col Ed Reed, 542 CTW, paper at 15th I/ITSEC

[3]Simulator flying hour cost determined by dividing CLS maintenance and support contract costs by yearly flying hours. Aircraft flying hour cost from HQ AMC

[4]Effectiveness of the C-130 WST, Nullmeyer, USAF Human Resources Lab, Paper at 14th I/ITSEC

pilots in the 'seat.' This essentially doubles the time the student has to practice training tasks. In the aircraft, one seat is always occupied by the instructor pilot; the second student 'dead heads' in the back. Finally, by moving training to the ATD's and freeing up flying hours, we were able to reduce the number of ramp aircraft required. Our analysis showed that we only needed four MH-53J aircraft to support our now reduced flying schedule, instead of the five that had been programmed. The 'freed' aircraft was returned to operational duty.

2. All on-aircraft training tasks preceded by simulation. Our process now involved three elements:

(a) Break all learning objectives down into the smallest possible tasks. For our purposes, the lowest task level was defined as concern for a single piece of equipment. For example, operate the INS.

(b) Teach tasks to proficiency in the least sophisticated ATD. The rationale for downloading aircraft time to simulation continued here to include downloading from higher to lower cost or sophisticated ATDs. We structured the training program to require students to be proficient in a single piece of equipment before proceeding in the syllabus.

(c) As the student progressed through the syllabus, related tasks were chunked together. These composite tasks were then trained to proficiency on the next higher level training device. On-aircraft training of any chunked task(s) only occurred after the student demonstrated mastery of the task(s) in simulation. Our desire here was to allow the student to use the limited on-aircraft time to develop task integration and crew coordination skills. A by-product was to increase flight safety. Prior to this, it was not uncommon for the instructor pilot and instructor flight engineer to help a student with a procedural task during night flight, only to realize that, while everyone had their heads in the cockpit pointing at and explaining switchology, no one was looking outside, flying the aircraft. With the new methodology, the student has mastered the procedural tasks and can concentrate on developing flying skills.

3. In order to support our training decisions, we realized that we had to acquire the highest fidelity training devices for each task taught at each level of simulation. The most obvious requirement driving this was correlated sensor training. We had to have a simulator that had real time out-the-window, FLIR, radar, and navigation system correlation. In addition, it had to have complete cockpit fidelity for switchology training. Without high fidelity or full correlation, all we would ever be able to do in the simulator was train part tasks, not integrated tasks. Part task trainers needed the highest fidelity possible to allow mastery of tasks and chunks of tasks.

4. Lacking the resources associated with many aircrew training systems, we had to save money. We decided to purchase off the shelf items whenever possible. For example, all of the pieces of our computer based training were all COTS items. If we needed something that wasn't commercially available, we either built it in-house (Helicopter Procedure Trainer) or contracted out (Training Observation Center), but always incorporated as many proven systems or sub-systems as GFE as we could to keep costs down.

5. Our final general procurement decision was to use lessons learned from each procurement to enhance our next. This is closely tied to our cost reduction measures. The development of our common Instructor Operating Station (IOS) is a good example. Originally a hard key IOS was acquired for the MH-53J WST. During the PTT acquisition, we upgraded the IOS to use touch screen technology. We used the software from the PTT IOS as the basis for the Operational Flight Trainer (OFT) IOS. After refining the OFT IOS, we went back and updated the WST and PTT. As a result, we have a common IOS across all systems.[5]

NEW AND MODIFIED TRAINING SYSTEMS

Armed with an implementation and acquisition strategy, we proceeded to build a new training course, saturated with high fidelity aircrew training devices at each level of training.

1. All of the existing HH-53 and new MH-53J courseware was moved from slide/tape or overhead projection medium to **Computer Based Training** (CBT). We are currently finishing our conversion to level 1 CBT. As soon as that is

[5] The same IOS was also used in procurement of the MH-60G WST and MH-60G OFT

complete, we will begin development of level 2 CBT. We made a number of CBT medium decisions up front, but three were particularly important. First, we built our courseware on 386-based computers. Although we had the resources to use 486 or SUN computers, our field units didn't. Since our ultimate goal was to copy all of our courseware and send it to every operational unit, we didn't want to develop courseware that the field couldn't use. We took the same approach in building graphics. Interactive video disk technology was probably affordable, we determined that data collection for construction and update of the courseware would be too intrusive and time consuming. Instead, we hired graphic artists and taught them to paint our graphics on computers (rather than teaching computer technicians to paint). Our final major decision was to build courseware that met the transportability requirements of the then draft Mil Std 1379D. Our primary reason is that we have three different contractors on site providing various levels of training to our fixed and rotary wing students. We saw a perfect opportunity to develop courseware on one contract and then 'give' it to ourselves for the other two.

2. Our first attempt at integrating CBT into the curriculum proved over zealous. Although our students for the most part like the CBT, they missed the interaction normally associated with a platform instructor. Our solution was to create **electronic classrooms** to allow instructors to use CBT in a lecture setting. The primary components of the electronic classroom are the computer aided podium (CAP) and a 35" television. The CAP contains a 386 computer that drives the courseware, a PC-VCR for adding live or simulator driven video, and software that integrates the two. In the Training Observation Center, we've also added a document camera to allow the instructor to use other non-electronic media. The electronic classroom provides a vehicle for CBT presentation to larger groups than the traditional one-on-one CBT approach. We are now modifying delivered CBT for use in electronic classrooms as group lecture material.

3. The **Helicopter Procedures Trainer** (HPT) is a static, dual purpose switchology and pre-flight trainer. For pilots, the cockpit of this converted USMC CH-53A is configured as a TH-53A using inert switches. In the contact and instrument phases of training, student can use the HPT to practice checklist procedures prior to or between rides in the TH-53A simulator and the aircraft. The back end of the HPT is configured as an MH-53J with operable ramp and doors. This allows student flight engineers and aerial gunners to practice loading troops, weapons, and vehicles.

4. The control display unit (CDU) is the primary interface between the crew and the integrated navigation system and is used for manual mission data entry into the mission computer. Mastery of the CDU is an essential prerequisite to all mission or sensor training. The **CDU trainers** are actual aircraft CDUs tied to 386-based computers through the serial ports. Software allows students to work through all data entry and update tasks. Six are available in the consolidated learning center, adjacent to the computer based training that teaches CDU operations. The CDU trainers provide a vehicle for students to acquire these critical skills while not tying up a WST or aircraft.

5. The **Pave Low Night Navigation Part Task Trainer** (PLNNPTT) is a high fidelity replica of the MH-53J cockpit. All switchology related to navigation, FLIR, moving map display, and radar is functional. A basic aero package and a small radar/FLIR database allow the crews to practice correlated sensor training including FLIR and radar offset and fly over updates and TF/TA flight. All modes of the TF/TA radar are functional.

6. The **TH-53A Operational Flight Trainer** (OFT) is a 6 degree of freedom, high fidelity simulator based on the TH-53A helicopter. It is a modification of an H-53 simulator and has a Compu-Scene V visual system that provides photo-specific a training environment. The TH-53A OFT is used for basic contact, emergency procedure, instrument, and initial day and night tactical (including night vision goggle) training. All cockpit switches and instruments are functional where required to support training tasks.

7. The **Integrated Electronic Combat Simulation System** (IECSS) is a replacement for the originally installed F-16 IEWTD. The IECSS has Air Force Electronic Warfare Center validated threat parametrics. It simulates not only tactical and strategic threats, but also multiple layers of threat system command and control. The system runs either independently (against one simulator)

or in a networked configuration. The IECSS was ready for training in October, 1993

8. The **MH-53J Weapon System Trainer/Mission Rehearsal System** (MH-53J WST/MRS) is a 6 degree of freedom, high fidelity weapons system trainer based on the MH-53J helicopter and is the cornerstone of our family of integrated training devices. It is a modification of the CH-3E simulator and, like the TH-53A OFT, also has a Compu-Scene V image generator (IG). Three aspects of the WST set it apart from most training simulators in use today. First, the WST has fully correlated FLIR, radar, navigation system, and out-the-window displays. The navigation system includes Global Positioning System (GPS), doppler, inertial navigation system (INS), control display units, and a central avionics computer that integrates the various flight systems. Because students are required to master correlated sensor operations, the WST needed to allow simultaneous heads-up and heads-down tasks to be performed. The second distinction of the WST is closely related to the first. The WST uses photo-specific, geospecific databases. Every database represents some real place in the world, not a fusion of various training environments. A by-product of this is that it set us up well for local and long haul networking. Finally, since most of our tactical training is at night, we need to have simulation that recreates the night environment. We decided to use wide angle colluminated windows as displays and modified light tables in the database, stretching the light tables to include night sensor phenomenon such as urban blooming. Students use the same night vision equipment in the simulator as they use on the flightline. When a student looks under the NVGs, he sees the correct representation of the environment - dark.

9. The **SOF-NET Intersimulator Network** (ISN) is our internal networking. All of our simulators (MH-53J, TH-53A, MH-60G WST, HC-130P WST, and MH-60G OFT), the IECSS, and the Training Observation Center will be on SOF-NET. This will allow us to fly formation, air refueling, and air combat maneuvering between simulators by just throwing a switch.

10. The **Training Observation Center** (TOC) is a 41 seat, multi-media auditorium with two primary functions (Figure 1). The TOC is an electronic classroom large enough to for all MH-53 students in all phases of training to be taught at one time. Since the TOC is cleared for classified material, it will soon be the home of our

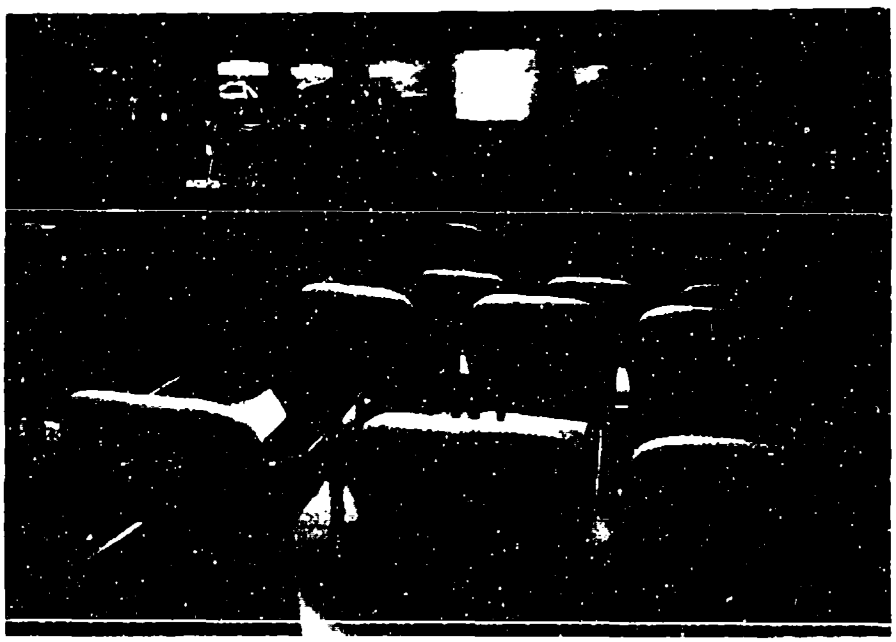


Figure 1. Training Observation Center

advanced tactical training. The second function of the TOC is as a simulator scenario control center. It serves as the pivot for the SOF-NET ISN. There are six role player stations, each with communication to all simulators. There is also a training director station which controls all activities in the TOC. The simulator display system (figure 1) has 10 screens. Each of the eight 35" displays is linked to a particular simulator. The training director can select any combination of OTW or sensor displays currently being used in the simulator for display on the screen. The large display on the right is home to a projected map of selectable scale that portrays all constructive and real simulation entities as icons. This allows the role players and training director to follow the action. The left large display is a repeater screen. During any phase of training, the training director can highlight the activities of a simulator by moving its' display signal to the large screen.

11. Both the TH-53A OFT and the MH-53J WST are equipped with video cameras that record all of the activity in the cockpit, including audio, FLIR, radar, and RWR presentations. The audio-visual recording studio captures the pilots' OTW visual. Taken together, these two tapes provide excellent **Mission Oriented Simulator Training**. We use these tapes during simulator sortie debriefs to reinforce good performance and refresh the memory of bad performance.

1993 MH-53J Training Course

The current MH-53J training course showcases the effect of high fidelity aircrew training devices integrated into a flight training curriculum. The course is longer now, primarily due to the increased sophistication of the MH-53J. More importantly, the course reflects the synergism available to trainers with access to advanced ATDs. Figure 2 is a slide developed for a briefing on the use of simulation in training. Their interest was the effective use of simulation prior to on-aircraft training. As you can see, every phase of MH-53J training follows the same general flow; academics, procedural or part task trainer(s), simulator, and on-aircraft. This flow, as discussed earlier, enables students to master component skills prior to bringing them together in the aircraft. Navigation system training is a good example of the new flow. Students receive lecture and then some CBT. They then proceed to the CDU trainer for relevant practice until they've mastered that task. Next is the PTT where they master the integration of the various navigation suite pieces. After the PTT, they practice in the WST. By the time they reach the aircraft, they've mastered use of the navigation system. Another change in the curriculum is the mix of time spent in simulator and PTT training to on-aircraft training. In the 1986 course, there were 7 hours of procedural training (preflight) and 28 hours of simulation, primarily concentrated in emergency, and instrument procedures. No tactical training was done in the simulator. The 1993 course shown in table 2, is heavily dependent on simulation. Although the course grew in length, the increase in flying hours (9.5) is quite small given the increase in training tasks. The real significance in the new program is the almost doubling of simulator hours and the types of tasks that are now trained in the WST. The course is almost a 50:50 mix of simulation and flight hours and, in the tactical and sensor phases, is actually much more heavily weighted in favor of simulation. The Pave Low phase, where integrated sensor operations are taught, is the most difficult part of MH-53J qualification. In the 1990 version of the course, this phase was 18 2.0 hour aircraft training sorties. The advanced capabilities of the WST, specifically the correlation of sensors, OTW visuals, and navigation system, has allowed us to restructure the phase to 12 simulation and 3 aircraft sorties. Advanced simulation has also allowed us to integrate new training tasks into the curriculum such as shipboard landing procedures and classified electronic warfare and evasive maneuvers.

THE RESULTS

The ultimate measure of effectiveness for any training program is how well graduates perform under operational conditions in the field. During Desert Shield, Desert Storm, and Provide Comfort, graduates were deployed directly to Southwest Asia. In one case, a graduate flew his first combat sortie three weeks after graduation. When asked about the quality of current graduates, a typical response from squadron commanders is; "I can't believe how good these guys are given the limited amount of flying time they get." The difference between our graduate's performance and their commander's expectations is a direct result of the quality and integration of

our ATD's. A second, more programmatic, measure of performance is the sustainability of the training program. During Desert Shield and Desert Storm, most aircraft mission spares, especially sensor and navigation systems, were sent to the Gulf. This is as it should be. However, the wing's training aircraft went without some equipment critical to training while still charged with graduating combat ready pilots. During the same time period (1990-92), we also took a 20 percent cut in MH-53J flying hours. By using the TH-53A aircraft for basic training and by increasing the reliance on simulation, especially the WST, we were able to graduate combat ready students on time (and in many instances early). The circumstances of the 90-92 time frame forced us to find a smarter way to train. We believe that advanced simulation is that smarter way.

LESSONS LEARNED

Advanced simulation works. It's an effective training platform, not only for procedural tasks but for crew coordination and complete mission training and rehearsal. Given the fiscal reality of diminishing training budgets (especially flying hours), more and more programs are going to look to advanced simulation as a cost effective means of combat crew qualification. To this end, here are three things to look for when making acquisition and integration decisions.

Correlation of displays. Most tactical aircraft or ground based weapon systems have advanced multi-mission sensors such as FLIR, radar, and targeting systems. They also have sophisticated navigation suites and, obviously, a large reliance on out-the-window cues. If a target in the FLIR doesn't appear in the targeting pod, out the window, or in the radar, students can't develop integrated sensor skills. Procedural trainers that teach switchology on a specific sub-system are a great start. However, a fully correlated WST is required to bridge the gap between the PTT and the weapon system. Without correlation, a WST is just an overpriced task trainer.

Databases. There are two different philosophies on database construction: either build a database of a piece of the real world or take pieces of the real world and put them together to create your own training world. The former are easy to describe; they correspond to someplace that really exists on the planet. The latter are really fictional planets, made up of a representative sample of every training environment the student could possibly use. Keep these issues in mind when deciding on databases.

(A) Maps, Charts, and Geodesy support. Students using real world databases can do mission planning using real DMA products, the same products they will use operationally. They can also use target area photography. Although it is possible to create maps of fictional planets,[6] timeliness, scale, and availability are concerns. It will also be difficult to get photography. Our goal has been to let students work with the materials in training that they will work with in the field.

(B) Round earth databases. Databases that assume a flat earth have two drawbacks. First, the earth is round. Placement of terrain and cultural features from a round earth to a flat database often result in displacement of features from their correct position, especially relative to other features. This increases the complexity of the sensor correlation issue. Second, line of sight calculations, either for terrain flight or threat occulting, will be incorrect given a flat earth database. This may or may not be a significant factor depending on the fidelity of other parts of the simulation. A grossly approximated threat parametric will probably give a bad detection solution on a flat or round earth database.

(C) Availability. Project 2851 offers a way to convert existing databases to a variety of formats, usable by a variety of image generators. This makes real world databases much more cost effective to acquire than in the past. SIF or GTDB capable image generators have access to hundreds of thousands of square miles of database today.

(D) Networking. Networked simulation is a requirement in many training programs. Crews are not usually employed alone; they shouldn't train alone. This requires that all players in the networked

---

[6]Database Correlatable Charts Enhance Simulation Training, Sherry Nathman, Hughes Training, Inc., Proceedings 14th Interservice/Industry Training Systems and Education Conference

simulation use a common "gaming" area. Recent efforts to combine virtual, real, and constructive simulation for multi-level, multi-layer training have all taken for granted the existence of a real world gaming area.

Training Transfer. This rather broad category deals with providing adequately sophisticated training devices for each stage of training to allow students to practice and then master tasks prior to proceeding to the next level of task complexity. Some programs may only need one training device. Others, like flight training programs, will need a full spectrum of vertically and horizontally integrated devices.

SUMMARY

We are hooked on simulation. Intermediate and lower complexity training devices have provided our students with the opportunity to master skills early and then practice them often. Advanced simulation has allowed us to train integrated sensor operations better and cheaper than in the aircraft. This has also allowed us to train tasks like shipboard operations and evasive maneuvering that we could only talk about before. Simulation has made our training, and subsequently our students, more effective. We see simulation as the training wave of the future. The greatest challenge for most programs is not whether to get involved in simulation, but to ensure that acquisition and integration decisions take advantage of mistakes and successes of others that already have.

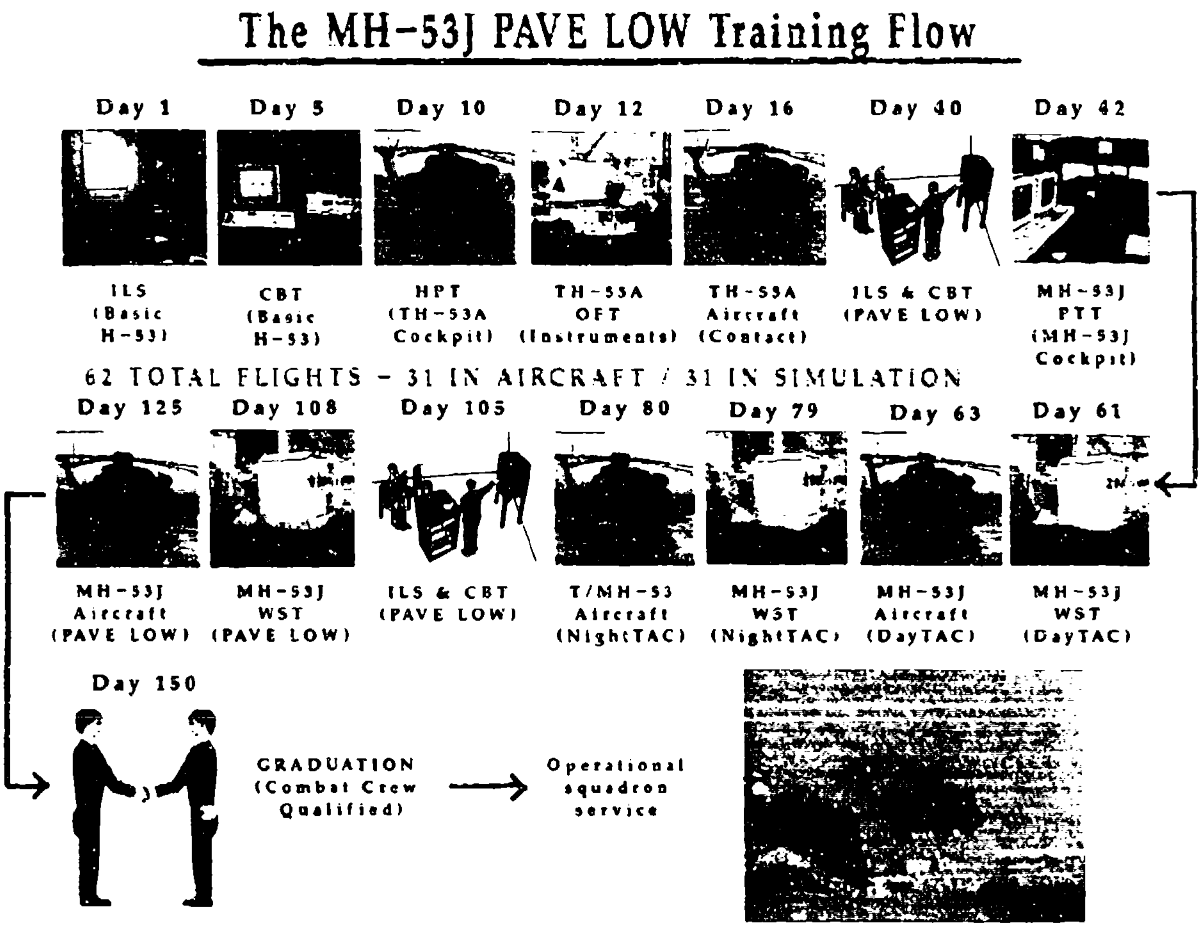


FIGURE 2. MH-53J Course Flow

Job Element	Academics	PT/PTT	'A' OFT	'J' WST	TH-53A	MH-53J	Total
Contact/EPs	57.5	8.0	8.0	4.0	10.8		88.3
Instrument	10.3		8.0				18.3
Remotes/Sling	23.7	2.0			5.4		31.1
Day Tac	18.0	2.0		5.0	2.0	10.5	37.5
Night Tac	4.0	2.0		5.0	5.0	8.0	24.0
Sensors	68.0	14.0		15.0			97.0
Pave Low		4.0		12.0		9.5	25.5
Evals	10.0				3.3	3.0	16.3
TOTAL	181.5	32.0	16.0	41.0	26.5	31.0	338.0

TABLE 2. 1993 H53P1 Course

THE ROLE OF THE MH-53J III E PAVE LOW WEAPON SYSTEM TRAINER/MISSION REHEARSAL SYSTEM (WST/MRS), IN PREPARING STUDENTS FOR OPERATION DESERT STORM, AND FUTURE OPERATIONS

Lt Col Russell Rakip
20th Special Operations Squadron
Hurlburt Field, FL

Mr. Jack Kelly, Ms. Sharon Appler and Mr. Peter Riley
Martin Marietta Corporation
Albuquerque, NM

ABSTRACT

The MH-53J Pave Low III E, WST/ MRS simulator was ready for training in August 1990 on the eve of the hostile actions in the Persian Gulf. It possesses the complete functionality of the aircraft and operates as both a trainer and a Mission Rehearsal Device. It graduated the first students to operational field units in the fall of 1990. This paper presents a survey of the MH-53J operators that are involved in real world special operations. The survey analysis evaluates how simulator training contributed to preparing this class, and future classes to meet the challenges of Desert Storm, and future operations. The survey captures the perceptions of training effectiveness from experienced operators tasked with performing missions along side these newly qualified crew members.

ABOUT THE AUTHORS

Lt Col Russell Rakip is the commander of the 20th Special Operations Squadron (SOS), Hurlburt Field, Florida. He has over 650 men and women in his command and has responsibility for both operations and maintenance for MH-53J Pave Low Special Operations helicopters. He previously served as Deputy for Special Operations of the 21st SOS, Woodbridge, England. He was the sole Air Force helicopter pilot involved in the Iranian hostage rescue attempt. He is a graduate of Lowell Technical Institute.

Mr. Jack Kelly is the Deputy Site Manager, for Contractor Logistics Support (CLS) with Martin Marietta (Formerly GE Aerospace) at Kirtland AFB, NM. He is responsible for the management and availability of four training devices, which include the MH-53J WST/MRS, the TH-53A OFT, the MH-60G WST/MRS, and the MH-53J PTT. He has flown the Pave Low as a flight examiner and is rated in eleven aircraft, both rotary and fixed wing. He holds a BS in Professional Aeronautics from Embry Riddle Aeronautical University.

Ms Sharon Appler is the manager of the Training Systems Support Center for the Kirtland AFB CLS. She has been with Martin Marietta for 11 years as a real-time software and systems engineer. As one of the original engineers on the MH-53J WST/MRS project, she performed the systems integration for the state-of-the-art systems simulated by the WST/MRS. Ms. Appler holds a BS in Mathematics from Wake Forest University.

Mr. Pete Riley is a Senior Training Analyst with the TSSC at Kirtland. He is responsible for the configuration management, quality assurance, and development of Computer Based Training. He has flown as a Pave Low instructor flight engineer while serving on active duty with the Air Force. He holds an AAS in Flight Engineering and is completing a BS in Vocational Education Studies with Southern Illinois University.

THE ROLE OF THE MH-53J WEAPON SYSTEM TRAINER/MISSION REHEARSAL SYSTEM (WST/MRS) IN PREPARING STUDENTS FOR OPERATION DESERT STORM, AND FUTURE OPERATIONS

Lt. Col. Russell Rakip
20th Special Operations Squadron
Hurlburt Field, FL

Mr. Jack Kelly, Ms. Sharon Appler, and Mr. Pete Riley
Martin Marietta Corporation
Albuquerque, NM

INTRODUCTION

The MH-53J Pave Low was selected by the commander of Allied Forces, General Schwarzkopf, as the first aircraft involved in hostile action in Desert Storm. After planning and rehearsing their critical raid, four MH-53J Pave Low pilots led Task Force Normandy into Iraq. Their elite joint force was comprised of their four USAF MH-53J Pave Low III "Enhanced" and eight Army AH-64 Apache helicopters. The task force split into two groups to attack a pair of Iraqi early warning radar stations. The strike was planned down to the last second. To open a corridor wide enough for Allied air forces, the sites had to be destroyed simultaneously. Both teams covertly flew at 80ft AGL, up shallow ravines, to within 3 NM of their target. Here, the Pave Lows marked the initial point, setting the stage for the Apache's to climb, form a right echelon and strike their targets with Hellfire missiles. By destroying two radar sites within seconds of each other, they opened a 20 mile wide gap in one of the world's toughest air defense networks. This allowed the Allied fighter and bomber forces unprecedented access to vital enemy command and control targets and virtually insured air superiority throughout the Gulf War.

The MH-53J Pave Low III helicopter is the most sophisticated rotary wing system in the world. It has state-of-the art systems that are second to none, developed for long range penetration behind enemy lines for both overt and clandestine operations. The highly task loaded Pave Low crews fly in all weather, day or night on Night Vision Goggles (NVG's). The MH-53J utilizes a sophisticated navigation suite consisting of Inertial Navigation System (INS), Doppler, Global Positioning System (GPS), Forward Looking Infra-Red (FLIR), and Terrain Following/Terrain Avoidance (TF/TA) radar. Along with the navigation systems the aircraft is equipped with an array of defensive systems, both automated and interactive. These combat crews stand ready to go, "ANY TIME, ANY PLACE," when only the best will do. Crews of the 20th Special Operations Squadron are among the most decorated veterans for Operation Desert Storm.

Early in the life of the MH-53J WST/MRS, soon after the first class of students graduated, word-of-mouth feedback on its benefits was reported back to Kirtland. This paper captures this feedback, and quantifies the benefits that the new simulator brought to the field operational units. The survey outlined in this paper was compiled from input by crew members formerly and currently assigned to the 20th SOS. Additional input came from crew members formerly and currently with the 21st SOS in England, as well as current and former crew members assigned to the 542d Crew Training Wing at Kirtland AFB in Albuquerque, NM. Squadron commanders who had command of these new crew members are also included in the survey. It is outside the scope of this paper to analyze or validate the curriculum being taught at the school.

This paper illustrates the significant benefits this state-of-the-art mission rehearsal system gives to the students who graduate to the Special Operations units.

THE SURVEY

The survey consisted of twelve questions asked of experienced Pave Low operators and their commanders. The questionnaire is attached as Appendix A. Each of the participants was asked to rate two groups of their peers. The groups were defined as:

Group 1 Crew members whom they flew with who were new to the Pave Low environment, trained between August of 1990 and June of 1992, or after March of 1993. This group of students graduated from the school when the simulator was an active and integral part of the curriculum.

Group 2 Crew members whom they flew with who graduated prior to August 1990, or after June 1992. The early waves of newly qualified crew members sent to Operation Desert Shield/Desert Storm came out of the school without the MH-53J WST/MRS. Between June of 1992 and March of 1993 the simulator was relocated, and a Block Modification Update was accomplished. The modification added new engine modeling, shipboard compatibility, avionics upgrades and other enhancements to bring the simulator up to the current, full MH-53J configuration. The simulator became ready for training again in March 1993.

Participants were asked to attempt to baseline in their mind each group of student crew members arriving at their operational units. This insures that the one particular individual who had exceptional natural talent, or someone with very little natural ability did not influence their answers.

The questions asked about both these groups of crew members were identical. Answers were then averaged, and the skills graded on a scale from 0 to 10. The results are illustrated in Figure 1 -- Survey Summary. The generalized results show that the skills rated by experienced crew members showed improvement, in some cases significant improvement, for the new crew who had the simulator as part of their curriculum. The survey did not scientifically measure the proficiency of the skills when new crew members perform their real world tasks. It does show that if you measure the perceptions of experienced crew members, the addition of the simulator did add significantly to the quality of the crews being sent into the field.

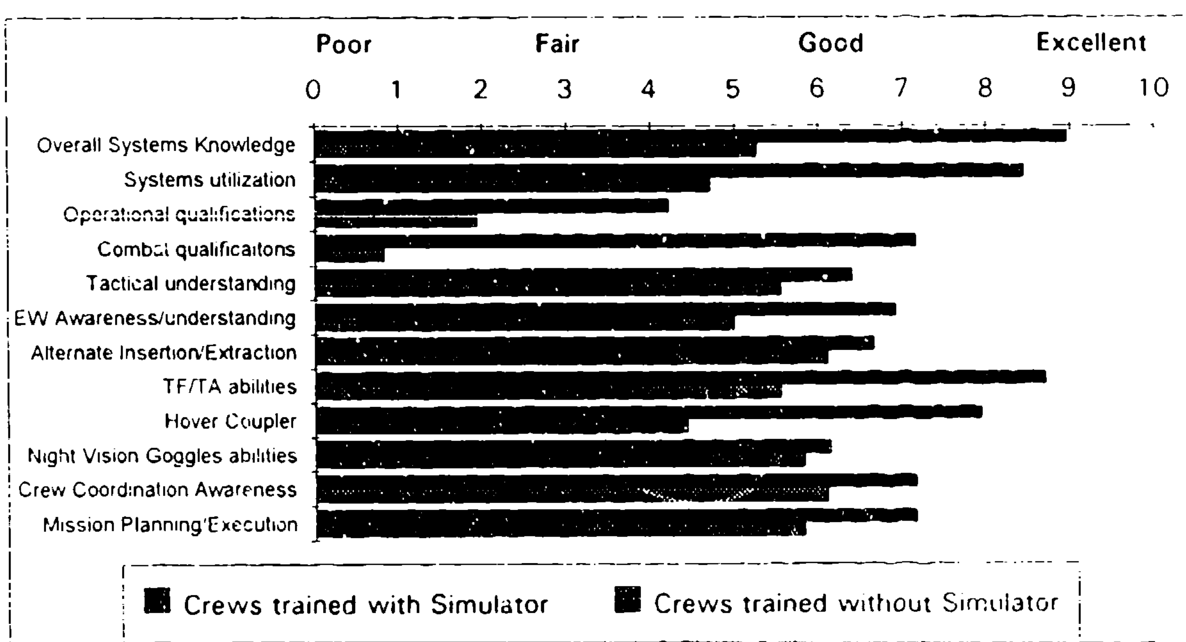


FIGURE I – SURVEY SUMMARY

SURVEY ANALYSIS

Overall Systems Knowledge

The Pave Low III E utilizes a sophisticated navigation suite consisting of Inertial, Doppler, and GPS navigation systems. The interface to the Enhanced Navigation Systems is through a Mission Computer Control Data Unit (ENS CDU), and the Doppler CDU. The systems have multiple redundant back up modes to insure that any single system failure does not adversely affect the Special Operations mission. The survey group felt that the students overall knowledge of the aircraft systems was good to excellent for the group who had the simulator available, vice a fair to good rating given to new crew who trained without the simulator. On a scale of 1 to 10, Group 1 scored almost a 9, while Group 2 scored close to a 5. This question represents the "book knowledge" of the systems that the students brought with them. It appears that the ability to assimilate the information in the book was significantly enhanced by the hands on practice available in the simulator. Survey participants currently serving as instructors at Kirtland commented that there was a dramatic increase in the number of students receiving "T"s (require further training) in the performance evaluations in the June 1992 to March 1993 time frame while the simulator was relocated and modified. This trend almost immediately reversed when the simulator came back on-line in March 1993.

System Utilization

This question asked the field operators to evaluate the hands-on systems skills that the new crew members brought with them to the field. The results of the survey show these skills are also in the excellent range (8.5 on a scale of 1 to 10) for those crews with a simulation curriculum. This is contrasted with the good to fair rating (4.5 on a scale of 1 to 10) of the crews who did not utilize the simulator. The previous squadron commander of the 20th SOS, deployed from Hurlburt Field in support of Operation Desert Shield/Desert Storm commented that the Flight Engineers were arriving on station at a quality level higher than he had previously observed. These engineers knew more about the systems and their operation than many of the experienced squadron instructors/evaluators already on station.

Operational Qualifications

Most of the survey participants had a difficult time answering this question. As they explained, the school concentrates more on the formal basics of operation. The operational units assume the hands-on aircraft mastery and operational training required for Special Operations crew. The ratings in this category ranged from ready to go through needing a lot of work to bring the newly qualfied crew member up to speed in Special Operations. The respondents volunteered that the students with a simulator background took approximately 2-3 months to bring them up to standards. The non-simulator crews took up to a year.

Combat Qualifications

Survey participants were asked to rate the combat capabilities of crew members being sent directly into the Persian Gulf war from Kirtland. Overall, crew members with a simulator background came highly qualified and in some cases they were almost immediately flying combat missions. According to survey participants with extensive experience in Desert Storm, some crew members were ready to fly combat missions with only two flights. The average additional flying time required for these newly qualified crew members was approximately 20 hours flying time. The average additional flying time required to get the non-simulator crews up to speed was 50-60 hours (at a minimum), with the general consensus that it could take in excess of six months.

Threat Avoidance

Along with the enhanced navigation capabilities of the Pave Low that bring it to a target on time, there is a suite of systems onboard the helicopter that allows Special Operators to arrive at their target undetected. Two questions in the survey asked about the ability of the new crew members to perform these functions. The survey shows that the awareness and understanding of the simulator-trained crew members of the Electronic Threat environment was enhanced when contrasted with those crew

members who did not use the simulator. The survey also shows that the crew member's ability to use the complex Terrain Following/Terrain Avoidance Radar system markedly improved by the use of the simulator. A contributing factor to their success is the full correlation of all the on-board systems in the simulator. The full field of view visual system (Figure 2) exactly correlates with the Forward Looking Infrared System, as well as the Digital Radar Land Mass System that simulates the TF/TA Radar. Electronic Warfare environments, Navigational Databases and onboard map display are also fully correlated.

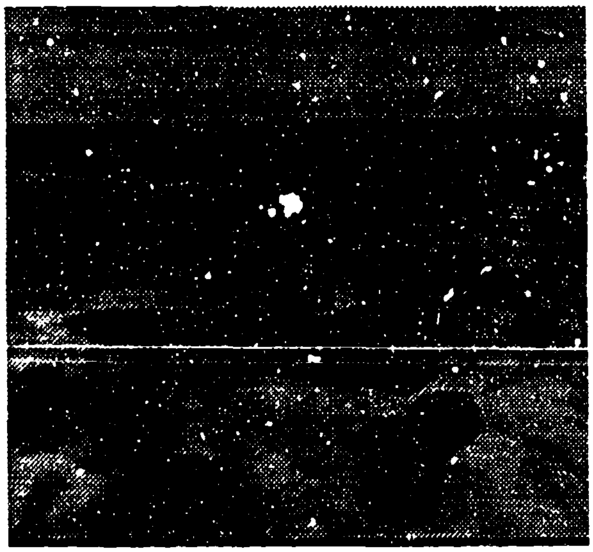


FIGURE 2 – WST/MRS VISUAL IMAGE

Tactics

Overall tactical skills taken from the school also benefited from use of the simulator. The tactical awareness and Alternate Insertion/Extraction skills improved between Group 1 and Group 2. Hover coupler skills were improved by nearly a factor of two, in those who had the opportunity to practice them in the simulator.

Crew Coordination

The next two areas of the survey do not show as dramatic an increase in capability in new crew members as in other areas. Crew coordination and Night Vision Goggle operation require a full Pave Low crew. The MH-53J WST/MRS currently simulates the pilot, co-pilot, and the primary flight engineer positions. The three additional crew members, the second flight engineer and two gunners/scanners in the doors in the aft cabin, are currently not included in the simulator. The most successful crews train to fly as they fight, as an integral unit, and their sense of cohesiveness is an important area that can be improved upon. Every survey participant commented that you can never have enough NVG training. For crews that fly in the dark of the night, improvement in the quality level of these skills is a lifeline. With the installation of the SOF-NET that will network the simulators in real-time, and with the addition of the gunner simulator to that network, the full crew compliment will be trained simultaneously, although in separate simulation devices. The quality of the training in these areas can certainly be brought up to the exceptional quality being produced in other areas of the simulation.

Mission Planning and Execution

One of the most important benefits the MH-53J WST/MRS brings to training is the ability to plan and fly a mission route in the simulator and then be able to perform your mission in real life. While crew members with simulator experience also did better in this category than their non-simulator counterparts, a greater benefit will be realized by the refresher students who return to Kirtland once a year from their operational units. All respondents to the survey were asked if they had attended refresher training at Kirtland both before and after the installation of the MH-53J. Those who attended both types of training responded unanimously that when the old unmodified HH-53C (SLICK) model simulator with no visual system was in use for refresher training, they thought the training benefits were minimal. The new simulator is perceived to be extremely beneficial for brushing up on the complex skills necessary to stay current in a highly tasked environment. Most participants believe that real world environments should be available to these crews to heighten their awareness and insert the real world, blood rushing, heart pumping, work-up-a-sweat fear of reality into their scenarios.

Additional Survey Comments

The one area that the crew members who trained exclusively in the aircraft brought with them was their ability to fly the aircraft. They had better "hands" from accumulated actual flight time. However, survey participants also noted that having someone who knew more about the complex capabilities of the Pave Low system was well worth the trade-off.

SUMMARY

This paper conclusively captures the often neglected views of the simulator system's "end user," the operational crews that fly with the new students being sent to field units around the world. This survey represents the perceptions of the seasoned captains and majors who flew into a war zone with a new pilot in the left seat and a flight engineer who went from school to war. Together these men had to get their aircraft to a specific target, on time. Simulators are traditionally used because of the considerable savings offered. They cost far less to operate than the real aircraft, and can be operated with no risk. In the opinion of the operators in the field, the MH-53J Pave Low III E, WST/MRS is providing superior, more qualified graduates. Student quality is higher than can be achieved through actual aircraft training. Commanders in the field were able to take new crew members and send them into combat shortly after arriving in-theater. This could not have been accomplished without the valuable training that the MH-53J WST/MRS provided. The instructors accomplish more sophisticated training and insert realism for the student with the simulator. Training is more effective than ever before. Crew members are arriving at operational units with higher skill levels than ever before. This is a direct result of simulating the state-of-the-art aircraft systems with a state-of-the-art simulator.

About the Simulator

The MH-53J WST/MRS was built and delivered by GE Aerospace for the Military Airlift Command. It was delivered in August of 1990, and declared ready for training. It is currently maintained under contract to Ogden OO/ALC by Martin Marietta Corporation for the 542d OPG/DOU in support of simulation training for the 551st and 542d training squadrons at Kirtland AFB, NM.

Appendix A --Survey Questionnaire

1. What level of overall Pave Low III E systems knowledge did the new crew members bring with them into the aircraft?
   - Excellent
   - Good
   - Fair
   - Poor
2. How would you evaluate their ability to utilize the aircraft systems?

3. Are the graduates of the schoolhouse operationally qualified to perform their missions, or do they require more training in the field?
   - Were ready to go
   - Need a little work
   - Need a lot of work
4. What level of tactical understanding did these crew bring with them?
   - Excellent
   - Good
   - Fair
   - Poor
5. What level of Electronic Warfare awareness and understanding did these crew bring with them?

6. What level of TF/TA skills did these crews bring with them?

7. What level of Hover Coupler skills did these crew bring with them?

8. What level of NVG expertise did these crew have?

9. How prepared were these crews to perform AIE (Alternate Insertion/Extraction)?

10. What level of crew coordination did these crew members have?

11. How were the crew's mission planning skills?

12. How close were the crew's to being combat qualified, and how much additional time was required to make the students operational once they arrived in-theater?

   Part I
   - Very qualified
   - Needed a minor amount of flying time
   - Needed an average amount of flying time
   - Needed a significant amount of flying time

   Part II
   - Additional Days required
   - Additional weeks required
   - Additional months required
   - In excess of six months required. (Note. participants defined this time period as up to one year.)

13. Additional comments?

Multiplayer Simulator Based Training for Air Combat

Major Steven C. Berger, USAF Peter M. Crane, PhD
Armstrong Laboratory, Aircrew Training Research Division
Williams AFB, AZ

ABSTRACT

Emerging simulation technologies provide new opportunities for training mission tasks and skills which have not been previously trained in simulators. Research is necessary to identify the tasks where additional training would most benefit mission ready pilots and air weapons controllers and which of these tasks represent training opportunities for networked simulators. Armstrong Laboratory, Aircrew Training Research Division has recently developed a SIMNET compatible network of F-15 cockpits with visual systems, an air weapons controller station, manned and digital threats, and an exercise control station. An evaluation of this system was conducted in which 23 F-15 pilots 13 air weapons controllers participated in simulated air combat exercises. Each team of lead pilot, wingman, and controller flew several offensive and defensive counter air missions against a force of up to six aircraft, anti-aircraft artillery, and surface to air missiles. Participants were asked to rate their interest in receiving additional training on each of 36 mission areas. After participating in the simulated air combat exercises, participants rated the value of the training received in the simulator system and the training currently received in their units for each of these mission areas. Data presented identifies, a) tasks which are of particular interest to aircrews, b) which tasks were better trained in the simulation system than in current unit training, and c) changes in pilot performance in simulated air combat related to levels of fighter experience.

ABOUT THE AUTHORS

MAJOR STEVEN C. BERGER is Chief of the Technology Development Branch assigned to the Aircrew Training Research Division of the Human Resources Directorate, Armstrong Laboratory, Williams AFB, AZ. He received his bachelor's degree in electrical engineering technology from Arizona State University in 1978 and has completed 33 hours toward the fulfillment of a master's degree in aeronautical science from Embry-Riddle Aeronautical University. His previous assignment was to Randolph AFB where he was Air Training Command Functional Check Flight Program manager, Chief, 12th Flying Training Wing Functional Check Flight Section, Squadron Executive Officer, 559th T-37 Pilot Instructor, 559 FTS. He currently manages both the Multiship Research and Development (MULTIRAD) program and Armstrong Laboratory's involvement with the Advanced Research Project Agency's WAR BREAKER effort.

PETER M. CRANE is a Research Psychologist with the Air Force Armstrong Laboratory, Human Resources Directorate, Aircrew Training Research Division at Williams AFB, Arizona. He received a BA in Psychology from Wittenberg University (1972) and an MA (1974) and PhD (1977) in Experimental Psychology from Miami University (Ohio). Before joining Armstrong Laboratory, Dr Crane was a Visiting Assistant Professor at Miami University--Middletown Campus and Assistant Professor of Psychology at the University of Pittsburgh at Johnstown. At Armstrong Laboratory, he has conducted research on the training effectiveness of simulated ground mapping radar imagery and simulated infrared imagery. Dr Crane's current research is focused on identifying the training opportunities and instructional strategies for networked, multiple aircraft simulations.

MULTIPLAYER SIMULATOR BASED TRAINING FOR AIR COMBAT

Peter M. Crane, PhD Major Steven C. Berger, USAF
Armstrong Laboratory, Aircrew Training Research Division
Williams Air Force Base, Arizona

INTRODUCTION

Background

Air combat requires a unique combination of perceptual, procedural, psychomotor, and cognitive skills. Dion and Bardeen (1990) point out that many existing ground-based trainers emphasize procedural and psychomotor skills by taking a part-task (PTT) approach. While the necessity of such training is unquestioned, Dion and Bardeen assert that, "Limited single-ship PTTs cannot provide multiship and integrative / multitask pilot experiences for team coordination skills,..." (p.467). Vraa (1990) further asserts that, "...the very best combat training flight environments, such as Red Flag, are necessarily constrained by factors relating to peacetime safety rules, resource availability, and funds,..." (p.459). These authors go on to state that multiplayer, air combat simulation against intelligent and responsive forces would fill in important increasing combat readiness. The emphasis of such training would be on cognitive and team skills such as mission planning, communication, tactics, attention management, decision making, and situational awareness.

Dion and Bardeen describe the components of such a training system and Vraa describes the characteristics of simulated opposing forces. The proposed training system would consist of 2-4 fighter cockpits integrated into a common battle space with manned or computer controlled threats. All appropriate systems would interact among players such as radar, weapons, and countermeasures. Houck, Thomas and Bell (1989) conducted a series of simulated air combat exercises using the simulator complex at McDonnell Aircraft Company. While these simulators were designed for engineering development, it was possible to reconfigure the system for multiplayer air combat training as Dion and Bardeen or Vraa propose. Pilots and air weapons controllers found that using the multiplayer simulation system provided effective training for:

a) Individual skills which can only be practiced infrequently such as radar sorting against multiple bogeys, and

b) Skills which require interaction with other players such as maintaining mutual support and working with an air weapons controller.

While the exercises conducted at McDonnell Aircraft Company demonstrated the value of multiplayer, simulator-based training for air combat, the simulation facility at McDonnell Aircraft was designed for engineering development and uses very high fidelity cockpits and mainframe computer technology. The Multiship Research and Development program (MULTIRAD) was initiated at the Aircrew Training Research Division in the Spring of 1991 to create a SIMNET compatible system of networked simulators for air combat training. The objective of MULTIRAD development was to integrate new and existing devices into a system which would provide high fidelity training for limited components of the F-15 air combat mission. The system would then be evaluated in a series of simulated air combat exercises known as the Training Requirements Utility Evaluation (TRUE). In the TRUE, teams of two F-15 pilots and an air weapons controller would either defend an air base against an attack or would escort a flight of F-16s attacking the air base.

Objectives

TRUE was conducted as an engineering evaluation of the MULTIRAD simulation system. In addition to engineering information, data were collected to identify mission tasks and skills which can be effectively trained using multiplayer simulation. Further, pilot performance in the simulated combat exercises

was analyzed to identify discriminators of mission success related to pilot level of experience.

RESEARCH METHODS

MULTIRAD Simulation System

MULTIRAD is a system of networked simulators designed to support air combat training which incorporates many of the features proposed by Vraa (1990) and by Dion and Bardeen (1990). MULTIRAD consists of two F-15C cockpits installed in wide field-of-view visual display systems integrated onto a SIMNET compatible network with:

a) Air Weapons Controller Station

b) Two (2) F-16 simulators configured to participate in TRUE exercises as Su-27 interceptors

c) Automated Threat System

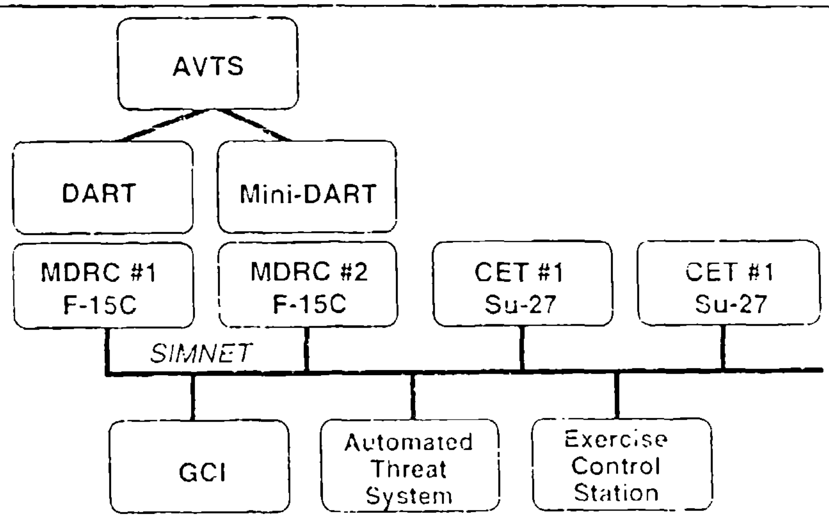
d) Exercise Control and Video Recording System

e) Independent Video Debriefing System

The layout of this system is shown in Figure 1. The simulators within MULTIRAD are limited fidelity systems in that many aircraft capabilities such as take-offs, refueling, emergency procedures, or landing are not simulated. High fidelity models, however, are used for functions critical to air combat. These functions include aircraft aerodynamics and performance, radars, missiles, and the effects of jamming and countermeasures. Platt and Crane (1993) present a detailed description of MULTIRAD.

TRUE Exercises

TRUE was conducted as a series of air combat training exercises similar to the exercises described by Houck et al. (1989). Three or four teams of lead and wing F-15 pilots plus an air weapons controller participated in four, week long exercises. During each of these weeks, teams flew seven, one-hour simulator sessions. During each session, teams flew three or four setups of either a Defensive (DCA) or Offensive (OCA) Counter-Air mission. On DCA missions, the team defended their home airbase against an attack from two MiG-27 fighter-bombers escorted by four Su-27 fighters (two manned and two computer generated). On OCA missions, the F-15s escorted a flight of four computer-generated F-16s attacking the air base which was defended by six Su-27 fighters, four computer-generated and two manned. Each DCA or OCA setup was initiated with the aircraft at 20,000' separated by 80 nautical miles. Computer-generated enemy force tactics were preprogrammed and the manned players



Su-27, MiG-27, SA-4,6,8

Figure 1. Multiship Research and Development (MULTIRAD) Network

followed a script during the early part of each engagement. There were six variations of enemy tactics for the OCA mission and seven for the DCA mission. In addition, the manned players were free to deviate from the script as circumstances demanded and the computer generated aircraft were programmed to defend themselves when attacked. Figures 2 and 3

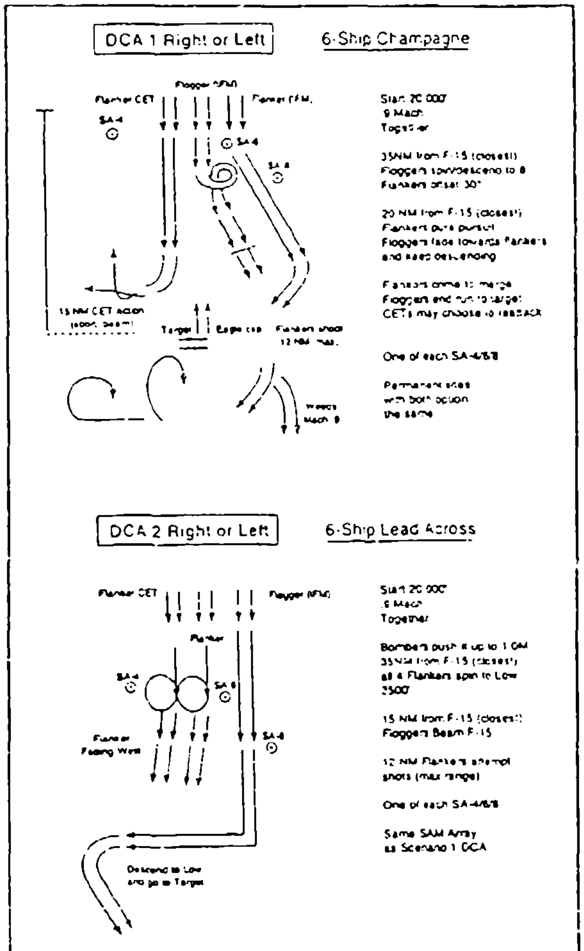


Figure 2. Two Variations of TRUE Defensive Counter Air (DCA) Mission

illustrate two variations of the DCA and OCA missions. F-15 pilots and controllers were not shown the enemy force plans nor were they informed about which variation would be seen on any setup. Variations were selected at random with the provision that no variation was repeated within a given simulator session.

At the beginning of each week, pilots and controllers were briefed about the TRUE objectives and procedures. Participants then filled out questionnaires listing 30 air combat mission tasks and skills and were asked to rate the desirability of receiving additional training on each. After a familiarization flight, teams flew three DCA and three OCA simulator sessions with three or four setups per session. Before each session, teams planned their missions

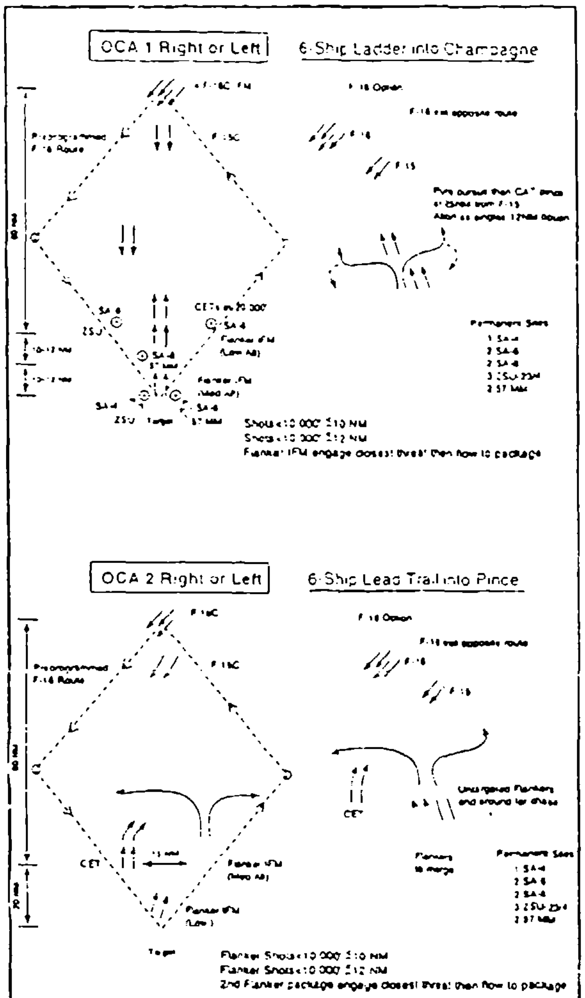


Figure 3. Two Variations of TRUE Offensive Counter Air (OCA) Mission

including call sign assignment, lookout responsibilities, and plans for attack, mutual support, and re-attack. Each engagement was video taped from the exercise control console using three, computer-controlled recorders. Each F-15 cockpit's front panel including the radar, radar warning, and armament control displays was recorded along with the overhead view of the engagement from the control console plus all audio communications and warnings. At the end of the simulator session, teams took the tapes to an independent

debriefing room which contained facilities for synchronized playback of the tapes. After debrief, each team's lead pilot completed a questionnaire describing any difficulties they experienced during the missions and lessons learned for the next simulator session. Participant comments and critiques were also solicited during daily meetings with all pilots and controllers. During some TRUE weeks, pilots flew one vs one engagements between the two F-15 cockpits. The results of these engagements are described in Crane (1993). After all simulator sessions had been completed, participants filled out a final questionnaire. On this questionnaire, participants rated the quality of training received in their current unit training program and from MULTIRAD for each of 30 tasks and skills.

TRUE Participants

Twenty-three, USAF, F-15 pilots and 13 air weapons controllers participated in TRUE exercises. In this paper, only the results from the pilots will be discussed. Pilot experience levels ranged from 300 to 2500 total flying hours with a median of 1400 total hours and 675 F-15 hours. The distribution of total flight hours of TRUE pilots by unit qualification is shown on Figure 4.

RESULTS

Interest in Additional Training

Pilots rated their interest in receiving additional training on each of 30 tasks using a scale from 1=additional training is not desirable to 5=additional training is extremely desirable. Mean ratings are presented in Figure 5. The 30 tasks on the figure are coded:

TACTICAL FRM	Tactical formation
VISUAL LOW	Visual low level flight
SEPARATION	Separation tactics
VISUAL ID	Visual identification of target aircraft
LOW ALT TAC.	Low altitude tactics
DEBRIEFING	Mission debriefing
EGRESS TAC.	Egress tactics
INTRAFLT COM	Flight communication
TWO SHIP TAC	Two-ship tactics
COM JAMMING	Communication jamming
TEWS	TEWS assessment
INTERCEPT	Tactical intercept
MISSILE	Missile employment
ECM	Employ ECM/ECCM
AI RESPONSE	Reaction to airborne interceptors
MUTUAL SUPT	Mutual support
WEATHER	All weather employment
CHAFF	Employ chaff/flares

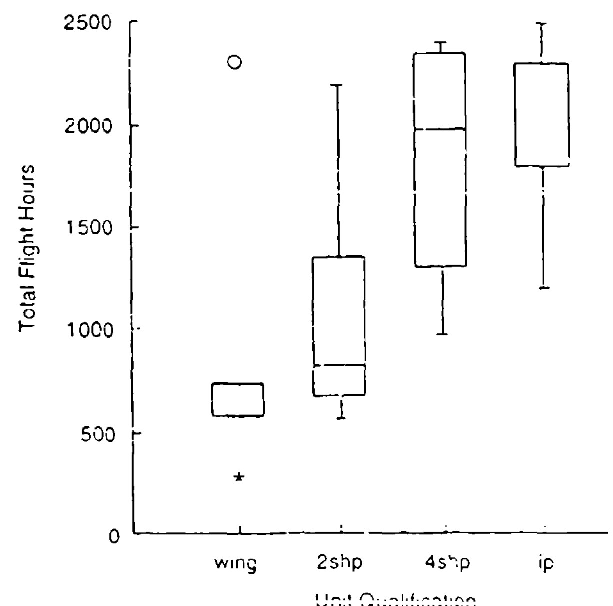


Figure 4. Distribution of TRUE pilot flight hours by unit qualification level (wing = mission ready wingman, 2shp = two-ship flight lead, 4shp = four-ship flight lead, IP = Instructor Pilot)

VISUAL LOOK	Visual lookout
EID	Electronic identification of target aircraft
TACTICS/PLAN	Tactics/mission planning and briefing
ESCORT TACTICS	Escort tactics
RADAR LOOK	Radar lookout
WORK W/ GCI	Work with AWACS/GCI
SAM DEFENSE	Reaction to surface-to-air missiles (SAMs)
RADAR SORT	Radar employment /sorting
BVR	Beyond visual range employment
ALL ASPECT D	All aspect defense
DACT	Dissimilar air combat training
FOUR+ BOGEYS	Multibogey, four or more

The five tasks with the lowest interest ratings are primarily visual tasks. The tasks with the highest rated interest in receiving additional training are tasks which can usually be practiced only in large exercises or cannot be practiced except in simulation, e.g. defense against SAMs. This result is in agreement with Houck et al. (1989) who found that pilots were most interested in receiving additional training for tasks which are least frequently practiced in the aircraft. Interest is low for tasks such as tactical formation or separation tactics which are practiced on all air-to-air sorties.

Value of MULTIRAD and Current Unit Training

Using the scale of 1=unacceptable to 5=excellent, pilots rated the value of their current unit training and MULTIRAD training for

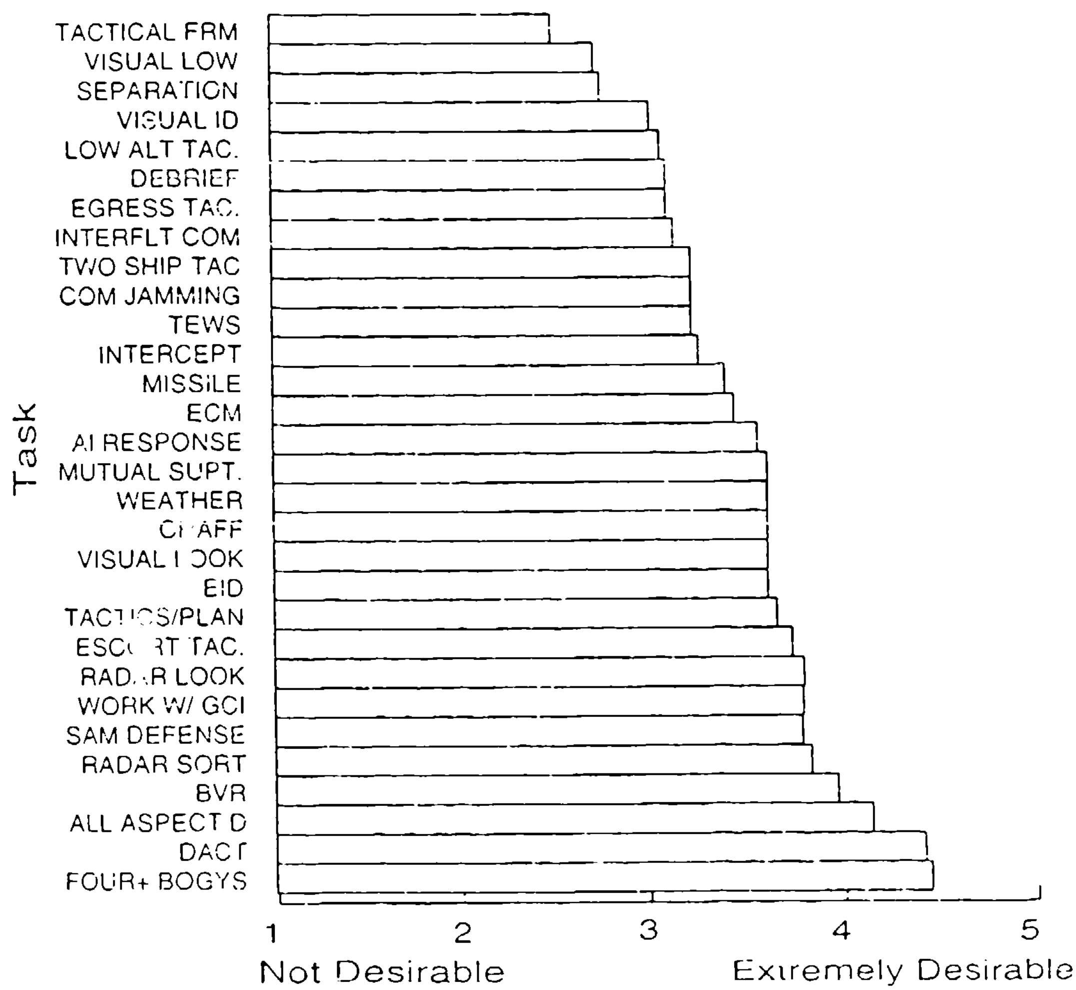


Figure 5. Pilot ratings of interest in receiving additional training for 30 air combat tasks

each of the 30 tasks. The lowest and highest rated tasks for unit and MULTIRAD training are:

	Current Unit	MULTIRAD
Lowest rated tasks	SAM defense	Visual id
	Com jamming	Tactical Form.
	ECM/ECCM	Com jamming
	Work w/ GCI	Visual lookout
	Four+ bogeys	Visual low alt.
Highest rated tasks	Visual lookout	Radar lookout
	Tactics /planning	Work w/GCI
	Mutual sup.	Radar sorting
	Tactical Form.	BVR employ.
	Radar lookout	Four+ bogeys

While some tasks, notably radar lookout, were rated highly for both MULTIRAD and current unit training, more tasks were given high ratings for one training environment and low ratings for the other. For example, visual lookout and tactical formation were the lowest rated tasks for MULTIRAD but among the highest rated tasks for current unit training. Likewise, work with GCI controllers and engagements against four or more bogeys were among the lowest rated tasks for current unit training and among the highest rated tasks for MULTIRAD. The differences between the mean ratings for unit and MULTIRAD training are plotted on Figure 6. Negative numbers indicate that unit training was rated higher than MULTIRAD training. Tasks with the highest ratings for additional training are printed in UPPERCASE type. Tasks which were rated

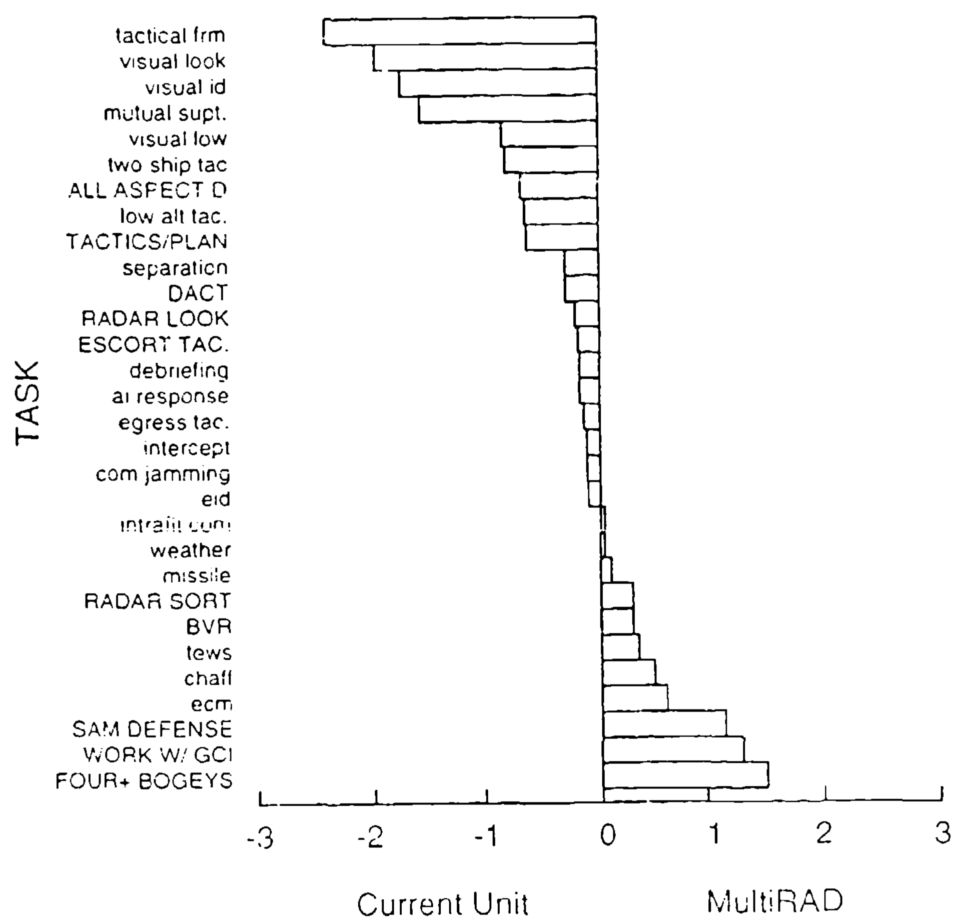


Figure 6. Differences between pilot ratings of training effectiveness for MULTIRAD and current unit training for 30 tasks; tasks for which there is high interest in receiving additional training are listed in UPPERCASE type

much higher in the current unit than in MULTIRAD are primarily visual. These tasks were also among the lowest rated tasks for interest in receiving additional training. Tasks which were rated higher in MULTIRAD than in the current unit are tasks which are not frequently practiced outside of large scale exercises. Also, tasks rated higher for MULTIRAD training were among the highest rated tasks for interest in receiving additional training. This finding is also inagreement with Houck et al. (1989) who reported that pilots rated multiplayer simulator based training higher than unit training for tasks which cannot be practiced in aircraft due to safety, cost, and security restrictions.

Pilot Comments

Pilot comments and critiques primarily fell into three categories: F-15 cockpit operations, simulation of opposing forces, and visual display problems. These comments were used to correct MULTIRAD deficiencies after each TRUE exercise (see Platt and Crane, 1993). Problems with the visual display systems, however, could not be corrected during TRUE. These problems are described in detail by Crane (1993). In brief, the pilots' major criticism was that aircraft were difficult to detect beyond 2 - 4 nautical miles and aircraft aspect or closure could not be determined beyond 0.5 - 1 nautical mile.

Mission Performance

A comparison of mission performance relative to experience levels was performed to determine what areas of mission success could be measured. Fighter pilot performance levels were extracted from previously video taped engagements. A total of over 267 OCA/DCA engagements were flown among the twenty-three pilots. Experienced versus inexperienced levels, rated by flight hours, were discerned as shown in Table 1.

TABLE 1

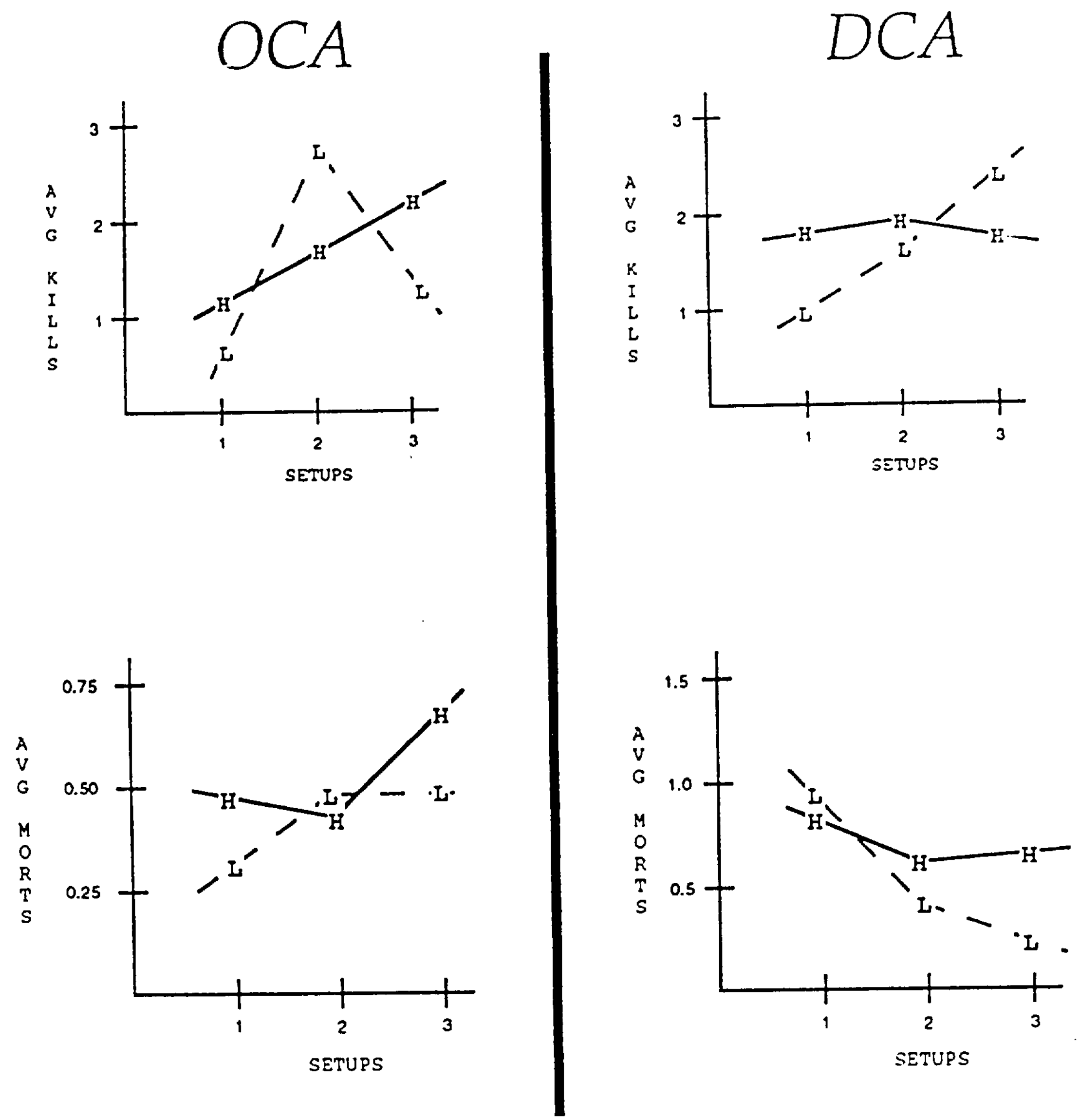
Inexperienced (Hours)	Experienced (Hours)
MR Wingman = 70 - 500	4-Shp Flt Ld = 600 - 1,000
2-Shp Flt Ld = 300 - 1,000	IP = 700 +

Data points collected included experience with live fire missile shots, experience in live mission exercises, number of various missile and gun shots taken during OCA and DCA scenarios, number of kills and misses, and mission success or failure. Week one through four data examined the F-15 team pilots flying against a full array of threats. The composite threat force on DCA's was 2 (F-15s) vs. 6 (MiGs) plus 3 SAM sites and on OCA's 2 (F-15s) + 4 (F-16 strike package) vs. 6 (MiGs) plus 3 SAM sites. Kill ratio involves two discriminators of mission success, average kills per setup and the number of times the pilot gets killed or morted. Kill ratio/exchange ratio can be examined for analysis of mission success. Most all engagements had the experienced pilots leading the formation and planning/directing the tactics. The exchange ratios were plotted and compared for the experienced and the inexperienced pilot as shown in Figure 7. The discriminator, average kills per setup, examined both OCA and DCA mission scenarios. Overall the complexity of the OCA scenario resulted in the inexperienced pilot to have large fluctuations of kills, however, this type of pilot did show improvement. The expert pilots performance improved more linearly. The DCA scenarios were not as complex, involving only combat air patrol and defensive lane protection. Therefore, results show an increase in kills for inexperienced pilots. Repeated setups for high time pilots showed no marked change. The other facet discriminator, mortality, examined the survival rate of the F-15 pilots. In the OCA missions the results for experienced pilots indicated a higher survival rate initially, however, since the experienced pilot is also the leader and responsible for wingman and strike package this could explain the higher mortality rate with increased setups. The complexity of the OCA for lead involves situational awareness and a host of other factors which are beyond the discussion of this paper. The survival of the wingman during OCAs show an increase mortality rate and then leveling out. This would seem to indicate a rapid learning curve for inexperienced pilots using such a training system. Again, since DCA missions are somewhat less complicated a significant learning curve is predominate in both categories of pilots. The ability to rapidly reset the engagements and learn from mistakes caused a faster learning

curve than can be currently trained in actual aircraft. Experienced pilots continued to improve but at a more linear rate. This could be attributed to adaptation and refinement of leads tactical game plan as the scenarios progressed.

DISCUSSION

TRUE was conducted as an engineering test of the MULTIRAD system. However, TRUE also provided the opportunity to gather



Legend

L = Inexperienced Fighter Pilots

H = Experienced Fighter Pilots

Figure 7. OCA and DCA Kill Ratio versus Experience / Inexperience

behavioral data on training interests and the potential applications of mulaship, simulator-based training for air combat. Overall, the results of TRUE are in agreement with the findings of Houck et al (1989). Although the engineering development simulator system used by Houck et al. at McDonnell Aircraft had more capability than the present MULTIRAD system, TRUE pilots rated simulator training higher than current unit training for a similar list of tasks. These include tasks which can be performed single-ship but are infrequently practiced such as radar sorting against multiple targets. The high rated tasks also include skills which can only be practiced within the context of a team. These tasks include work with a GCI controller and operations against four or more bogeys.

Dion and Bardeen (1990) predict that the major benefit of multiplayer combat training will be the development of team skills. These benefits will come from, "training situations in which the team learns to develop adaptive teamwork skills required in real time for uncertain problems, and an expanded repertoire of team experiences," (p.468). It is unclear, however, whether the benefits of multiplayer simulation come from development of team skills or from the development of individual skills at tasks which can only be practiced within a team context. Crane (1992) predicts that training will be of most benefit for skills which pilots have had the least opportunity to practice. This prediction is based on cognitive models of expertise which assert that a journeyman lacks the expert's extensive knowledge base and is unable to quickly recognize the significant elements within a mission and to select an appropriate response. Simulator based training can provide the foundations for building an expert's knowledge base. Since the tasks in TRUE which were highly rated for MULTIRAD training are both team tasks and infrequently practiced, no conclusion can be drawn regarding Dion and Bardeen's prediction that the benefit of multiplayer simulation will be development of team skills.

Low time fighter pilots certainly benefit from a multiplayer simulation. Mission success criteria depends on pilot performance. Certainly part of this performance is the pilots ability to maintain a high kill ratio level. It seems that this is a good gauge or metric when determining his abilities in combat. It is so important that many fighter squadrons implement a TOP GUN program to track this performance. The discriminators which make up kill ratio, kills versus mortality, certainly indicate that inexperienced pilots will gain more, i.e. have a much more accelerated learning curve, when exposed to near realistic training scenarios over a repetitive period of time. Many other factors also influence mission success; situational awareness, weapons employment ability to name a few. All of these factors comprise the kill ratio metric. Simulation plays a key role for the low time pilot. It is important to them to be exposed to all the pitfalls of flying in combat. By doing so you increase the knowledge base at a more rapid rate utilizing those precious few training sorties for items that cannot be trained in the simulator. The results prove that a high fidelity multiplayer simulation system improves fighter skills.

CONCLUSIONS

The results of TRUE support the contention that multiplayer simulator based training is a valuable training medium for increasing wartime readiness especially for less experienced pilots. Multiplayer simulation is best suited for continuation training as an adjunct to existing unit training. Current training practices are best suited for training many of the tasks required of air combat pilots. Training in simulator systems similar to MULTIRAD is best suited for training tasks which cannot be practiced in aircraft due to cost, safety, and security restrictions.

REFERENCES

Crane, P. (1993). Evaluation of Two Wide Field-of-View Display Systems for Air Combat Training. Presented at: Euro Display '93: International Display Conference, Strasbourg, France.

Crane, P. (1992). Theories of Expertise as Models for Understanding Situation Awareness. Presented at: 13th Psychology in the Department of Defense Symposium, Colorado Springs, CO.

Dion, D. and Bardeen, I. (1990). Multiship Air Combat Team Mission Trainer (TMT) Concept: Meeting a Tough Continuation Training Requirement. Proceedings of 12th Interservice/Industry Training Systems and Education Conference. Orlando, FL: National Security Industrial Association.

Houck, M., Thomas, G., & Bell, H. (1989). The Training Potential of Multiplayer Air Combat Simulation. Proceedings of the Human Factors Society 33rd Annual Meeting. Santa Monica, CA: The Human Factors Society

Platt, P. and Crane, P. (1993). Development, Test and Evaluation of a Multiship Simulation System for Air Combat Training. Proceedings of 15th Interservice/Industry Training Systems and Education Conference. Orlando, FL: National Security Industrial Association

Vraa, R. (1990). Effective Air Combat Team Performance - With Bandit's Help. Proceedings of 12th Interservice/Industry Training Systems and Education Conference Orlando, FL: National Security Industrial Association.

NON-DIAGNOSTIC INTELLIGENT TUTORING SYSTEMS

Leo Gugerty and Kimberly Hicks
Mei Technology Corporation
San Antonio, Texas

ABSTRACT

The keystones of traditional intelligent tutoring systems (ITSs) have been complex procedures for student diagnosis and adaptive instruction based on diagnostic data. While some of these systems have been shown to be effective, they are also very expensive to develop. This paper describes another class of ITSs, non diagnostic ITSs, which do little or no student diagnosis, and concentrate their intelligence in other areas. Intelligent features of non-diagnostic ITSs include: modeling of experts' reasoning processes and cognitive representations (often using graphic displays), comparison of student and expert performance, and replays and summaries of student performance. While traditional, diagnostic ITSs are usually intended to be used in a stand-alone fashion, non-diagnostic tutors are designed to facilitate collaborative learning among students and between teachers and students.

The non-diagnostic approach to ITS development offers either a low-cost alternative to traditional ITSs or a way to expand the educational capabilities of traditional systems. This paper presents a framework for comparing the features of non-diagnostic and diagnostic tutors. A number of non-diagnostic and diagnostic ITSs are described, and data on the costs and educational effectiveness of each type of ITS is presented. Finally, obstacles to wider use of non-diagnostic ITSs are discussed.

ABOUT THE AUTHORS

Dr. Leo Gugerty is a cognitive psychologist in the Training Technology Division of Mei Technology Corporation. His research focuses on complex problem-solving skills (such as troubleshooting) and effective interventions to train these skills. He has a Ph.D. in Cognitive/Experimental Psychology from the University of Michigan, Ann Arbor, and has held research positions at Educational Testing Service, Lockheed Engineering and Sciences Co., and the University of Houston.

Kimberly Hicks is an Industrial Psychology Specialist in the Training Technology Division of Mei Technology Corporation. Her work is concentrated in the areas of training systems analysis, and survey development and analysis. She has an M.A. in Industrial Psychology from St. Mary's University, San Antonio, TX.

NON-DIAGNOSTIC INTELLIGENT TUTORING SYSTEMS

Leo Gugerty and Kimberly Hicks
Mei Technology Corporation
San Antonio, Texas

INTRODUCTION

There is a sense in which the goals of traditional intelligent tutoring systems (ITSs) are both too ambitious and too narrow. Most "classical" (or traditional) ITSs, such as the LISP Tutor (Anderson & Reiser, 1985), are designed to provide tutoring in a stand-alone setting (i.e., without a human teacher present). This ambitious goal requires that the ITS handle all aspects of the very difficult task of tutoring, including expert problem solving, student diagnosis, tailoring instruction to changing student needs, and providing an instructional environment (e.g., a simulation). On the other hand, the goal of developing very intelligent stand-alone ITSs is narrow in the sense that it limits our conception of how intelligence can be incorporated into computer-based training and education. One key problem with focusing on stand-alone ITSs is that we may overlook intelligent computer-based systems that include the teacher as part of the tutorial interaction.

ITSs are currently being developed that break with the pattern of traditional ITSs. An example is the Intelligent Conduct of Fire Trainer (INCOFT), an ITS to train the skill of identifying aircraft from radar displays (Newman, Grignetti, Gross & Massey, 1989). INCOFT does little student modeling and relies on a teacher to provide much of the instruction. Its intelligence lies in its ability to advise students when their performance differs from an expert's, to model experts' reasoning for the student, and to provide useful summaries and replays of student performance that can be discussed by the student and the teacher. Thus, INCOFT acts as an intelligent teacher's aid, and facilitates collaborative learning.

The goal of this paper is to describe ITSs like INCOFT, and compare these to traditional ITSs like the LISP Tutor. The key features that differentiate INCOFT and the LISP Tutor are student diagnosis and adaptive instruction. The LISP Tutor performs student (or cognitive) diagnosis; that is, it makes inferences about the knowledge and misconceptions underlying student performance. Having a detailed student diagnosis enables the LISP Tutor to adapt its instruction to small changes in student knowledge during a tutoring session. INCOFT, on the other hand, simply records student performance without making inferences about it. Therefore, INCOFT must rely on the teacher to adapt instruction to fine-grained changes in student knowledge. Whether or not an ITS performs student diagnosis has a large effect on how it can be used in instruction. Therefore, we will refer to systems like the LISP Tutor as *diagnostic* ITSs. This term is intended as shorthand for a system that performs both detailed diagnosis and adaptive instruction based on the diagnosis. Systems like INCOFT will be referred to as *non-diagnostic* ITSs (and sometimes as intelligent teacher's aids).

There are a number of reasons for exploring non-diagnostic ITSs. The first reason concerns the difficulty of the tutoring task, as described above. Recently, some ITS researchers have suggested that some tutoring tasks, such as student diagnosis, will require long-term basic research before solutions are found (Burger & DeSoi, 1992). A second reason is that augmenting a teacher's knowledge with a non-diagnostic intelligent teacher's aid may provide just as much and perhaps more educational benefit as replacing a teacher (or at least part of a teacher's task) with a stand-alone, diagnostic ITS. A third reason is that traditional diagnostic ITSs are very expensive to develop and are applicable only in narrow domains. Non-diagnostic tutors can cut the cost of tutor development by

eliminating the need for some of the complex components of traditional ITSs.

In this paper, we will describe specific features, advantages, and disadvantages of both non-diagnostic and diagnostic ITSs, and estimate the cost of each approach in terms of type and level of development work. This overview should allow someone considering developing or using an ITS to understand the costs and benefits of each approach.

COMPARISON OF DIAGNOSTIC AND NON-DIAGNOSTIC ITSs

Table 1 contains a list of some of the key capabilities or features of ITSs. The table is organized in terms of the four components of traditional ITSs, the expert, diagnostic, instructional, and interface modules. For each capability in the table, a range of options is presented, from "high-tech" options that rely on the computer to perform the pedagogical function (e.g., diagnosis), to "low-tech" options that rely on the teacher (or other students) to perform the function. The table shows the capabilities of two diagnostic ITSs, the LISP Tutor and Sherlock I (Lajoie & Lesgold, 1989), and two non-diagnostic ITSs, INCOFT and the Maintenance Aid Computer Hawk Intelligent Institutional Instructor (MACH III) (Kurland, Granville & MacLaughlin, 1992). The LISP Tutor has primarily high-tech features. Sherlock I is less sophisticated than the LISP Tutor, but still possesses the essential capabilities of a diagnostic ITS. The two non-diagnostic ITSs have primarily low-tech features, except in the case of their interfaces, which use high-tech features such as realistic simulation and modeling expert reasoning and representations.

A few points should be made about Table 1. First, the terms high-tech and low-tech are not meant to connote a value judgment. Complex technology is not always the best technology. Second, there is not a strict correspondence between diagnostic ITSs and high-tech features, on the one hand, and non-diagnostic ITSs and low-tech features, on the other. The two non-diagnostic ITSs use some high-tech features, as the table shows. Also, diagnostic ITSs can incorporate low-tech features that facilitate teacher involvement, as do non-diagnostic systems. An example of this is the Sherlock II maintenance skills tutor, which provides precise student diagnosis and adaptive instruction as well as feedback (such as replays and summaries of student performance) intended to foster collaborative learning (Katz & Lesgold, in press). Sherlock II can be considered a hybrid of a diagnostic and a non-diagnostic ITS.

In the following, we first describe each of the ITS capabilities in the table and explain the differences between low-tech and high-tech options. Then, some of the diagnostic and non-diagnostic ITSs in Table 1 are compared in terms of the table features.

Comparison of ITS Features

Expert Module - As Table 1 notes, an important question concerning the expert module is whether it simulates human thought processes. *Black-box* expert modules solve problems using methods completely unlike humans, while *glass-box* experts attempt to simulate the important human thought processes used in the task being instructed (Burton & Brown, 1982). The LISP tutor is an example of a glass-box expert module. This tutor uses hundreds of production (if-then) rules to represent the knowledge and strategies used in LISP programming in a detailed manner. INCOFT uses a black-box expert (mathematical equations) to solve aircraft identification problems.

The main advantage of a glass-box model is that its detailed model of human thought processes allows it to more specifically and accurately diagnose student knowledge and misconceptions, and then base instruction (e.g., explanations) on specific student weaknesses. The main disadvantage of glass-box models is their cost. The expert module for the LISP tutor is based on Anderson's ACT* theory of human learning and problem-solving, which is based on years of research and theoretical work (Anderson, 1983).

A second important question characterizing expert modules is whether they generate the steps to solving problems online, when presented with a brief problem description, or have the specific solution steps pre-stored in their memory. A system that generates problem solutions online usually can solve a wider variety of problems than a system that relies on "canned" (pre-stored)

Table 1. Capabilities of Diagnostic and Non-Diagnostic Intelligent Tutoring Systems

Key: Diagnostic Systems: LISP Tutor (L), Sherlock I (S); Non-Diagnostic Systems: INCOFT (I), MACH III (M)

ITS Module Capabilities	Low-Tech					High-Tech	
Expert Module							
Simulates human thought processes?	no expert module			black-box expert	I,M,S	glass-box expert	L
Generates problem solutions online?	no expert module			canned solutions for pre-selected problems, or generated solutions based on a simple algorithm	I,M,S	solutions generated for a wide variety of domain problems	L
Diagnostic Module							
Info in student model	no student model	performance history	I,M	specific knowledge & skills based on a black-box expert (e.g., issue-based tutoring)	S	specific knowledge, skills, and misconceptions based on a glass-box expert (e.g., model tracing)	L
Use of student model	no student model	performance replays and summaries	I,M			to choose problem/topic sequence, knowledge/skills to coach, and level of coaching	L,S
Instructional Module							
How is the content and sequence of topics/problems determined?	by teacher		I,M	preset by tutor developers		generated online	L,S
How is the method of instruction determined?	by teacher		I,M	preset by tutor developers	L,S	generated online	
How is the content and timing of instructional interventions determined?	by teacher		I,M	preset by tutor developers	L,S	generated online	L,S
Group or individual use?	collaborative		I,M	stand alone	L,S	both	
Interface							
Simulates real-world task context?	not at all			to some extent	M,S	closely	I,L
Models expert reasoning and representations?	very little			somewhat	L,S	extensively	I,M

problem solutions. The LISP Tutor can generate problem solutions online, while Sherlock I and MACH III use pre-stored problem solutions. INCOFT generates solutions online using a very simple algorithm.

To use pre-stored problem solutions, one must conduct a task analysis, which is time consuming and requires some specialized knowledge. However, this is a less difficult task than developing a model of problem solving that can generate solutions to arbitrary problems in a domain. Thus, the use of task analysis and pre-stored problem solutions is less expensive and more widely applicable than developing a system that can generate solutions online.

Diagnostic Module - The second major component of an ITS, the diagnostic module, allows the system to create student models that record aspects of individual students' performance and knowledge. The ITS then uses information in the student model to tailor its instruction to the needs of individual students.

The most advanced diagnostic modules use performance data concerning the actions students take during problem solving and/or the final results of their problem solving to make inferences about the knowledge and skills behind individual students' performance. A powerful method for making these inferences, called *model tracing*, can be used if an ITS has a glass-box expert module that models human thinking. Model tracing is used in the LISP tutor. As the student uses the computer to plan and write computer programs, the diagnostic module matches each problem solving action taken by the student with the specific knowledge (i.e., production rules) that the expert module would use to produce that action. The diagnostic module also contains production rules to represent specific student misconceptions, so that when students make errors, it can match them with the underlying misconception. The diagnostic module can then record in the student model production rules that a student knows well, rules the student knows less well, and misconceptions.

A slightly less sophisticated method of student diagnosis is *issue-based tutoring* (Burton & Brown 1982). An issue-based tutor makes inferences about the knowledge underlying student performance, like a model-tracing tutor. However, issue-based diagnosis can be accomplished with a black-box expert module, whereas model-tracing requires a glass-box expert. Sherlock I uses a complex version of issue-based tutoring.

The information in a student model created by model tracing or issue-based diagnosis can be used by the instructional module in a number of ways, such as in determining the contents of hints and explanations and in selecting problems for students. For example, if the diagnostic module infers that a student mistake is based on knowledge the student knows fairly well, the instruction module can give only a general hint to the student. On the other hand, if the student mistake is based on knowledge the student knows poorly, the instruction module can give a detailed explanation of the mistake and the correct move.

The least sophisticated diagnostic modules record only data about student performance in a student model, making no inferences about the knowledge underlying this performance. An example is INCOFT, which monitors and records the aircraft-identification actions students take while they observe radar displays. It also records the timing of students' actions. These data are used by the instructional module in two ways: to provide replays of a problem in which differences between the student's and an expert's performance are pointed out; and to create summaries of student and expert performance on a problem. INCOFT does not use its limited student model to adjust the instruction based on a student's performance or to select problems. These tasks are left up to the teacher.

In the extreme case, some computer-based training systems record no student performance data, and do no student diagnosis.

Since the student model created by model tracing relies on a glass-box expert that closely simulates human thought procedures, this kind of diagnostic capability is expensive and time consuming to develop. The minimal student model used by INCOFT is obviously much easier to develop. The effort required to develop issue-based student models varies widely depending on

the complexity of the issues and the complexity of the schemes by which issues are updated.

Instructional Module - The third major component of an ITS is the instructional module. Planning and delivering tutorial instruction is a complex, interactive task. The decisions a tutor must make include: 1) curricular decisions regarding the content and sequencing of topics or problems, and 2) instructional decisions regarding the type of instructional intervention, the content and timing of instructional interventions, and the overall method of instruction. The tutor must choose from a variety of instructional interventions, such as exposition (e.g., explanations, examples of concepts, modeling of procedures), coaching (e.g., hints and explanations during problem solving), and asking and answering questions. Methods of instruction also vary widely, including direct instruction, guided discovery learning, and Socratic dialog. In addition, the advantage, and challenge, of tutorial interaction is that all of these decisions can be changed frequently based on the tutor's assessment of the student's progress, motivation, and learning style.

As Table 1 shows, an ITS can make these curricular and instructional decisions online (during a tutorial interaction) using a comparison of the student's and the expert's knowledge states. Alternatively, an ITS's developers could make some or all of these decisions on a one-time basis and hardwire these decisions into the ITS's algorithm. Finally the ITS could leave curricular/instructional decisions up to the teacher.

The first curricular/instructional decision shown in Table 1 focuses on curricular decisions, such as problem sequencing. The LISP Tutor and Sherlock I choose problems online based on diagnosis of individual students' knowledge. At the other extreme, INCOFT requires the teacher to make these decisions. The next instructional decision shown in the table concerns the overall methods of instruction. Almost all existing diagnostic ITSs have a single method of instruction that is used consistently. For example, the LISP tutor uses a directive, problem-based method of instruction, with immediate feedback after errors. Non-diagnostic ITSs like INCOFT and MACH III rely on the teacher to determine the method of instruction.

In terms of more specific decisions about the content and timing of instructional interventions, most diagnostic ITSs make some decisions online, while other decisions are preset by the developers, as is shown in the table. For example, in the LISP Tutor, the content of specific hints and explanations was preset by the developers. However, the tutor makes a number of instructional decisions online, such as when to intervene (based on student errors), whether to provide a general hint or a detailed explanation (based on the number of student errors or the student's request), and the topic of the hint or explanation (based on the diagnostic module's assessment of the missing knowledge or misconception underlying the student's error). Again, with non-diagnostic ITSs, the teacher must decide what instructional interventions to use, for example, how to use the replays and summaries of student performance. Finally, Table 1 also characterizes the instructional modules of ITSs according to whether they focus on collaborative or stand-alone use.

To summarize this subsection, even though the LISP tutor is one of the most intelligent of ITSs, the instructional decisions that it generates online are based on fairly simple algorithms. This is typical of other diagnostic ITSs. Much of the intelligence of the LISP tutor, and most diagnostic ITSs, lies in the diagnostic and expert modules. For most ITSs, many important curricular/ instructional decisions, such as the overall instructional method and the content of explanations, are made on a one time basis by the system developer and cannot be changed by the tutor itself during operation or by the teacher. Developing ITSs that can flexibly make difficult curricular/instructional decisions is a long term research goal. The approach taken by non-diagnostic ITSs is to leave these difficult decisions up to the teacher.

Human-Computer Interface - The final ITS component in Table 1 is the interface. Two factors that distinguish low-tech and high-tech interfaces are whether the tutor simulates the real-world task context, and whether the interface allows students to use experts' reasoning and knowledge representations while using the tutor.

Non-diagnostic tutors concentrate their intelligence in the interface.

A realistic simulation environment can help students transfer knowledge from the tutorial to a job situation. INCOFT uses a realistic, simulated radar display that allows students to solve aircraft identification problems in real time. The artificial intelligence and psychological expertise required to build a realistic simulation often is less extensive than that needed to create glass-box expert and diagnostic modules. On the other hand, expertise in computer graphics and video is needed, and a thorough task analysis must be done. The LISP Tutor also uses a realistic interface.

Bonar (1991) has suggested that the effectiveness of an ITS will be greatly improved if the interface allows students to see and work with experts' reasoning and representations while solving problems. For example, MACH III represents expert troubleshooting knowledge in terms of "troubleshooting trees", which show all the general and specific faults that could cause a particular symptom in a radar system, as well as the troubleshooting tests to conduct for each specific fault. Students can view the appropriate tree on the computer in order to understand why a particular test was recommended. As in constructing a simulation environment, extensive artificial-intelligence knowledge is not required to build an interface like this. Rather, one needs a careful analysis of experts' problem solving processes for the task to be tutored.

Comparison of Particular Diagnostic and Non-Diagnostic ITSs

To summarize the discussion of the capabilities of diagnostic and non-diagnostic ITSs, we will compare the capabilities and the effectiveness of the LISP Tutor with those of INCOFT and MACH III. As the table shows, the LISP Tutor is a diagnostic ITS. It uses high-tech approaches in its expert and diagnostic module, that is, a glass-box expert and model tracing. Its instructional module uses a mixture of high-tech techniques (e.g., choosing the topics and level of detail of hints and explanations online) and some less sophisticated ones (using only a single, preset method of instruction). The LISP Tutor's interface simulates real-world programming interfaces closely, and sometimes allows students to use expert task representations (e.g., by showing students templates of LISP functions to fill in).

INCOFT takes a non-diagnostic approach. Although its expert module generates problem solutions online, it does this using a simple algorithm. INCOFT's diagnostic module records only performance data about the nature and timing of student responses. The tutor's instructional output consists of replays and summaries of students' performance, and demonstrations of expert performance. These were designed to be used more as informational aids for teachers and students than as stand-alone instructional interventions. INCOFT leaves the decision about how to use these aids, and most other curricular/instructional decisions, up to the teacher. The strength of INCOFT lies in allowing students to practice a real-time task on a realistic interface, and then, via replays and summaries, providing students with comparisons of their performance and that of experts. The important instruction with INCOFT occurs when the teacher and student (or groups of students) discuss the student's replayed problems. While using the replays and summaries, students do not have the pressure of real-time performance, and can evaluate and discuss their performance.

So far in this paper, we have examined the differing capabilities of diagnostic and non-diagnostic ITSs, and given a rough indication of the cost or level of effort these capabilities require to develop. Diagnostic tutors are more sophisticated in how they model students' knowledge states and adapt instruction to students' needs. Non-diagnostic tutors focus their intelligence on modeling experts' task knowledge and providing replays and summaries that can facilitate collaborative instruction and learning. Because of the difficulty of developing student diagnosis schemes, diagnostic tutors are usually more costly to develop. A key question for someone who is contemplating developing an ITS is whether the added sophistication of diagnostic ITSs is worth the cost. To begin to answer this, we will present some data on the effectiveness of diagnostic and non-diagnostic tutors.

The example we have used for a diagnostic tutor has been the LISP tutor. The model-tracing approach used in this tutor has also been used in tutors for geometry, algebra, and calculus (Merrill,

Reiser, Ranney, & Trafton, 1992). Anderson and Reiser (1985) found that students using the LISP tutor took 15.0 hours to complete a set of programming exercises, much faster than students who completed the exercises on their own (26.5 hours), and almost as fast as human-tutored students (11.4 hours). Each group performed equally well on posttests of their programming knowledge. Other model-tracing ITSs, such as Anderson's Geometry Tutor and the Graphical-Instruction-In-LISP (GIL) Tutor, have also been found to be more effective than traditional instruction (Merrill et al., 1992).

Two points should be made about these findings. First, in all of these evaluation studies, students using ITSs also received classroom instruction from a teacher. Thus these studies suggest that model-tracing tutors are effective in outside-the-classroom situations. The studies do not suggest that these tutors can replace human teachers altogether.

The second point concerning the effectiveness of model-tracing tutors is that some of this effectiveness may be due to other aspects of the tutors besides model-tracing, such as the structured editor in the LISP tutor and the graphic interfaces in the Geometry Tutor and GIL. However, studies have shown that when both model-tracing diagnosis and its associated instructional guidance are removed from the LISP tutor and GIL, students learn slower and sometimes perform worse than with the full versions of these tutors (Corbett & Anderson, 1991; Merrill et al., 1992). Another study compared a version of GIL that provided very little instructional feedback (that is, where model tracing diagnoses were used only to point out when students made errors) to versions that gave more detailed explanations of the locations of and reasons for errors (Merrill et al., 1992). The versions with more detailed instructional feedback resulted in faster and better student learning. These studies suggest that each of the key aspects of intelligence in a model-tracing ITS -- model tracing diagnoses and adaptive instructional feedback based on these diagnoses -- can lead to increments in students' learning.

On the other hand, the intelligent capabilities of model-tracing tutors do not always lead to better learning. For example, students using the version of GIL without model-tracing diagnoses or instructional feedback performed better on a debugging posttest than students using the full-fledged GIL (Merrill et al., 1992). Thus, the detailed and immediate feedback characteristic of model-tracing tutors may deprive students of the opportunity to make, and learn to correct, errors. This issue deserves further investigation, since debugging is an important aspect of programming skill.

Few studies have been conducted on the effectiveness of non-diagnostic tutors, as these tutors have been developed more recently than diagnostic tutors. INCOFT was not evaluated formally. However, a controlled study of the effectiveness of MACH III has been completed (Acchione-Noel, Saia, Williams & Sarli, 1990). MACH III was developed by the same company as INCOFT, and shares its focus on classroom-based learning via replays, summaries, and graphic representations of expert knowledge (e.g., troubleshooting trees). In keeping with MACH III's intended use as a classroom teaching aid, the study compared the use of this ITS with the traditional classroom methods of practicing troubleshooting in a radar maintenance class. The traditional methods involved using procedure manuals and schematics (paper-based practice). Both the MACH III and the "paper-based" group also received lectures and practice on the actual radar equipment.

Although the MACH III students did not perform any better than the paper-based group on practical and written troubleshooting posttests, the tutor students did perform more consistently (i.e., with lower variability). Also, the MACH III group solved significantly more, and more difficult, troubleshooting problems during the class than the paper-based group.

The lack of significant differences in student posttest performance in this initial study should not be taken as a general criticism of non-diagnostic tutors, for a number of reasons. First, the instructors felt they needed more training on how to use MACH III in the classroom. Second, the instructors tended not to use some of the more advanced features of MACH III, such as the troubleshooting trees, because they thought these gave students too much help. The Army school where MACH III was tested (Ft. Bliss) has continued to use the tutor in classes following the tests (Kurland et al., 1992).

Because MACH III was used in a different instructional context, the evaluation of this ITS cannot be compared easily to the evaluations of diagnostic tutors. The MACH III evaluation studied tutor use in the classroom and used a control group that received extensive classroom instruction. The studies of diagnostic ITSs (e.g., Anderson & Reiser, 1985; Lajoie & Lesgold, 1989) looked at stand-alone ITS use, and found that students using these ITSs performed better than students who received no additional instruction.

CONCLUSION

Non-diagnostic ITSs offer a potentially fruitful approach to computer-based education and training that complements the approach taken by traditional diagnostic ITSs. The lack of student diagnosis in non-diagnostic tutors will likely result in lower tutor development costs. In addition, the non-diagnostic approach promises to have positive educational value. Non-diagnostic features such as modeling experts' representations, replaying and summarizing students' performance, and focusing on collaborative learning implement some of the key aspects of the successful cognitive-apprenticeship approach to training and education (Collins, Brown & Newman, 1989).

The non-diagnostic approach can be applied in the development of new ITSs and in converting existing systems (e.g., expert systems) to ITSs. Many expert systems use black-box expert modules. Black-box modules are difficult to convert to a traditional ITS because the expert problem-solving knowledge in the expert system does not mimic human knowledge and thus is not in a form that can be easily used for student diagnosis (Anderson, 1988). However, if one is developing a non-diagnostic ITS that does little or no student modeling, then the conversion task will be much easier. For example, the Air Force's Armstrong Laboratory is developing a computer-based training system based on the Integrated Maintenance Information System (IMIS) (Link, Von Holle & Mason, 1987). IMIS is a job aid that will assist flightline maintenance technicians repair aircraft. IMIS contains an expert system that gives troubleshooting advice to technicians. Since this expert system is closer to a black-box than a glass-box system, non-diagnostic ITSs are being used as models in developing the training version of IMIS. Preliminary design work suggests that non-diagnostic features can be added to IMIS to create a relatively low-cost training system, because extensive task analysis will not be necessary beyond that performed for developing the expert system.

Before non-diagnostic ITSs can become widely used, however, a number of obstacles must be overcome. The first obstacle concerns how to integrate these ITSs into the classroom and train teachers to use them. The importance of considering these issues was highlighted by the instructors in the MACH III study, who asked for more ITS training and tailored the ITS use to their instructional goals in ways not intended by the developers.

The second obstacle to widespread use of non-diagnostic ITSs is the lack of empirical validation of their effectiveness. Conducting rigorous research that tests these systems in their intended educational settings (i.e., classrooms and other collaborative learning situations) should be a high priority.

Acknowledgments: - We would like to thank Valerie Shute, Patricia Hsieh, Ron Sheffer, William Walsh, Kevin Singley, and Elizabeth Gibson for comments on the paper.

REFERENCES

Acchione-Noel, S., Saia, F.E., Williams, L. & Sarli, G. (1990). Maintenance aid computer hawk intelligent institutional instructor: Training developments study. (Report No. TRAC-WSMR-TEA-90-051). White Sands Missile Range, NM: U.S. Army TRADOC Analysis Command.

Anderson, J. R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.

Anderson, J. R. (1988). The expert module. In M. C. Polson & J. J. Richardson (Eds.), *Foundations of intelligent tutoring systems* (pp. 21-54). Hillsdale, NJ: Erlbaum.

Anderson, J. R., & Reiser, B. J. (April, 1985). The LISP tutor. *Byte*, pp. 159-175.

Bonar, J. G. (1991). Interface architectures for intelligent tutoring systems. In H. Burns, J. W. Parlett, & C. L. Redfield (Eds.), *Intelligent*

tutoring systems: Evolutions in design. Hillsdale, NJ: Erlbaum.

Burger, M. L., & DeSoi, J. F. (1992). The cognitive apprenticeship analogue: A strategy for using ITS technology for the delivery of instruction and as a research tool for the study of teaching and learning. *International Journal of Man-Machine Studies, 36*, 775-795.

Burton, R. R., & Brown, J. S. (1982). An investigation of computer coaching for informal learning activities. In D. Sleeman & J. S. Brown (Eds.), *Intelligent tutoring systems.* New York, NY: Academic Press.

Collins, A., Brown, J. S. & Newman, S. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. Resnick (Ed.), *Knowing, learning and instruction.* Hillsdale, NJ: Erlbaum.

Corbett, A. & Anderson, J. (1991). Feedback control and learning to program with the CMU Lisp Tutor. Paper presented at the 72nd Annual Meeting of the American Educational Reseach Association, Chicago, IL.

Kurland, L., Granville, R. & MacLaughlin, D. (1992). Design, development, and implementation of an intelligent tutoring system for training radar mechanics to troubleshoot. In M. Farr & J. Psotka (Eds.), *Intelligent instruction by computer.* Philadelphia, PA: Taylor & Francis.

Lajoie, S. P & Lesgold, A. (1989). Apprenticeship training in the workplace: Computer-coached practice environment as a new form of apprenticeship. *Machine-Mediated Learning, 3*, 7-28.

Link, W. R., Von Holle, J. C., & Mason, D. (1987). Integrated Maintenance Information System (IMIS): A maintenance information delivery concept. (Report No. AFHRL-TP-87-27). Dayton, Ohio: Wright-Patterson AFB.

Merrill, D., Reiser, B., Ranney, M. & Trafton, J. G. (1992). Effective tutoring techniques: A comparison of human tutors and intelligent tutoring systems. *The Journal of the Learning Sciences.*

Newman, D., Grignetti, M., Gross, M., & Massey, L. D. (1989). Intelligent Conduct of Fire Trainer. Intelligent technology applied to simulator-based training. *Machine-Mediated Learning, 3*, 29-39.

ADVANCES IN LEARNING AND INSTRUCTIONAL DESIGN THEORIES

Katharine C. Golas, Ph.D.
Southwest Research Institute
San Antonio, Texas

Conrad G. Bills
Loral Defense Systems
Akron, Ohio

ABSTRACT

Learning and instructional design theory is the body of principles proposed by psychologists and educators to explain how people acquire skills, knowledge, and attitudes. Learning theory is used in formal instruction to facilitate and accelerate the learning process. When applied to the practice of instruction, learning principles derived from theories can guide the instructional designer in improving the effectiveness and efficiency of the learning activities of a program. This discussion of learning theory is an attempt to express the human process of learning in terms that can be applied in training and education. The categories of human activity have been delineated by learning theorists. This paper uses those categories to establish a framework for how learning takes place and addresses how learning theory is applied to the selection of instructional strategies as well as the media selected to deliver the instruction. The paper also addresses the fact how, in real life, the various types of learning are integrated. This integration of human activities is discussed in terms of schemas, enterprise theory and metaskills.

ABOUT THE AUTHORS

Dr. Katharine C. Golas is manager of the Instructional Systems Section at Southwest Research Institute. She began her career in ISD in 1977, by using the Interservice Procedures for Instructional Systems Development Model to develop print-based exportable job training packages. During the past 16 years, she has directed over 75 ISD projects, including 20 interactive videodisc projects and 10 Digital Video Interactive (DVI)®* projects. She is currently directing research and development efforts using advanced multimedia training technologies. In 1992, she led a project team to redesign the Air Force ISD model and methodology. She has a Ph.D. and M.A. in Instructional Systems from Florida State University.

Conrad G. Bills is senior training systems analyst for Simulation and Training Engineering, Loral Defense Systems. He has 20 years of experience with the U.S. Air Force operations and support programs in training and education, instructional system development (ISD), evaluation and administration, scientific management, and applied psychology. During the past seven years with the Air Force, he was senior training systems analyst for the Training Systems Product Group and Support Systems Engineering at the Aeronautical Systems Center, Wright-Patterson AFB, Ohio. He directed the contract for updating the Air Force ISD process, including the first volume for application of ISD in training system acquisition. He has an M.S. and B.S. in counseling and psychology/mathematics education from Brigham Young University.

---

*®DVI is a registered trademark of Intel Corporation.

ADVANCES IN LEARNING AND INSTRUCTIONAL DESIGN THEORIES

Katharine C. Golas, Ph.D.
Southwest Research Institute
San Antonio, Texas

Conrad G. Bills
Loral Defense Systems
Akron, Ohio

INTRODUCTION

The human process of learning is complex. The methods for preparing instruction have grown out of an attempt to understand this complex process. Irrespective of how well instruction agrees with the human process of learning, instruction will continue to take place. People will be taught or they will find out on their own what they need to know and how to do what they need to do in order to meet the demands placed upon them. In the course of events, unforeseen detours will be made, unneeded costs will be expended, and in some cases there will be unnecessary failures. A body of psychologists and educators have attempted to better understand this complex process of human learning and derive methods for improving the effectiveness and efficiency of corresponding instruction.

Categories of Learning

Early attempts to understand the complexities of learning resulted in classifications in categories of learning. Bloom[1] was the first to classify domains of learning (cognitive, affective, and psychomotor. His classifications helped to teach others about learning, but the transition of this taxonomy of learning to instructional design was not realistic. Gagné[2] was able to define a classification which corresponded to learning phases. Gagné's five categories are: intellectual skills, verbal information, cognitive strategies, motor skills, and attitudes. In preparing a classification for educational technology applications, Merrill[3] classified learning domains in the performance-context matrix (remember, use, and find; fact concept, procedure and principle. Reigeluth[4] provided another sort by types of learning including memorization, understanding, skills application, general skills, and affective learning. In general, these types of learning are grouped as psychomotor or behavioral, intellectual or cognitive, and feeling or affective.

Behavioral learning activities can be simple, such as stimulus-response-reinforcement event[5], or more complex, such as riding a bicycle. The behavioral approach in instructional design is to select learning outcomes which demonstrate that the desired learning has taken place, often referred to as the terminal behavior[6]. Behaviors are directly observable and measurable. The application of this approach to instructional design works well with specific tasks that have corresponding terminal behaviors. It is more difficult to apply in more complex goals involving multiple objectives and an understanding of a person's thought process and associated feelings[7].

Cognitive learning activities draw attention to mental process and methods for organizing information or experiences or establishing relationships. Understanding a cognitive approach is more difficult because the activity is unseen and the descriptions are often abstract. Knowledge is not directly observable but it is measurable. The application to instructional design allows another dimension for explaining expected learning outcomes.

The **affective domain** is the emotional context for learning, a dimension for which little theory has been derived. The affective domain, such as attitudes and motivation, is also abstract and internalized. In situations where there is little behavioral feedback and no verbalized response, understanding the affective domain is quite difficult. The application to instructional design is therefore more difficult. There is an understanding that motivation is important and attitude, particularly in critical situations, must be instilled. Making the affective domain a part of instructional design established importance for gaining the attention of

the learner, establishing relevance for the learner, and building the learner's confidence in and satisfaction with the instruction[8].

Integration in Application

Categories of learning help us understand the complex process of human learning, but in application the approach is to integrate. Gagné reflected on his categorization of learning types, and apologized for causing the segregation of the categories[9]. He noted that this separation was only to assist in understanding the complexities of learning. He never intended for segregation, but rather integration in application. He said that in reality all the categories play together as a whole. This means in application to instructional design, the approach should be a comprehensive one, incorporating all categories of learning. Gagné and Merrill[7] proposed that such an integration of multiple objectives be conceived in terms of the pursuit of a comprehensive purpose in which the learner is engaged, called enterprise.

COGNITIVE THEORY

More contemporary attempts to define the complex process of human learning are known as cognitive theories. These theories focus on what is going on inside the learner's mind. Two respected models of cognitive theory are the information processing model and the social interaction model.

Information Processing Model

The information processing model says that the learner's brain has internal structures that select and process incoming material, store and retrieve it, use it to produce behavior, and receive and process feedback on the results. A number of cognitive processes are involved in learning, including the "executive" functions of recognizing expectancies, planning and monitoring performance, encoding and chunking information, and producing internal and external responses.

The purpose of instruction is to activate the internal processes in order to facilitate the acquisition of new skills, knowledge and attitudes. Different kinds of learning outcomes require different means for activating the internal processes, and these means are called instructional strategies. Application of the information processing model in instructional design is shown in Table 1. This table shows the relationship between learning processes and phases and provides examples of general instructional design strategies that support learning.

Social Interaction Model

Social interaction theory says that learning and the consequent changes in behavior take place as a result of interaction between the learner and the environment. This environment can facilitate or inhibit learning. A context promoting cooperative learning is more helpful to students than one promoting individual work, especially with regard to learning attitudes. An effective application is when the interaction among students is given support through tutoring and feedback, such as peer tutoring.

Social interaction states that students learn as they confront the response demands built into the activities in which they participate. Good activities are built around the attainment of multiple goals even within a given activity. The application to instructional design is the formation of activities which engage students in active forms of learning. These active forms of learning help develop values as well as critical thinking skills, are built around "important" content, and are well matched to the learner's abilities and interests. The instructional delivery approaches are varied in order to facilitate the attainment of the multiple goals.

INTEGRATION OF HUMAN ACTIVITIES

Cognitive theories help us understand the complexities of human learning. Tennyson[10] notes that, in reality, the cognitive systems are dynamic, they interact, and they are integrated. The relationship of integration in development is depicted by the progress from the apprentice to the novice and then the expert. The apprentice develops to a point of initial proficiency. Integrative goals have been obtained complete to terminal objectives. Achievement of the expected learned capability occurs in the defined condition or environment at the desired standard or level of activity. An assessment is made that the integrative goals have been reached. The performance is now that of the novice, able to do the job but still needing"aging and experiencing." From novice, development continues over time. Skills become more robust, ability to adapt among variations is expanded, choices leading to solutions are more appropriate

Table 1. Application of Information Processing Model in Instructional Design

Learning Process	Learning Phase	Instructional Aim	Strategies to Support the Processes	Examples
Expectancy	Motivation	Build relevancy and communicate the goal	• Set the stage • Personalize the context • Create uncertainty	• Tell a story • Provide a demonstration • Ask leading questions
Perception	Apprehending	Focus attention	• Use novel or interesting examples • Activate the learner's senses	• Use color • Use print techniques such as bold face type or italics • Introduce sounds, smells, real objects, video
Working Storage	Acquisition	Present information in manageable units	• Organize the content • Produce a visual image that illustrates abstract information	• Use mnemonics • Chunk information • Outline information • Use imaging techniques (such as concept mapping and Information Mapping®)
Encoding	Processing	Build upon existing knowledge	• Put content into meaningful context	• Provide analogy, metaphor, simile • Provide meaningful examples and nonexamples
Storage	Retention	Merge new information with existing knowledge	• Encourage rehearsal • Provide for spaced review	• Create new examples • Paraphrase information • Have learner verbalize new SKA
Retrieval	Recall	Attach the new skill, knowledge, attitude (SKA) to environmental cues	• Provide situations in which new information should be used	• Practice applications of new SKA • Have learner teach new SKA
Validation of Understanding	Feedback	Test accuracy of new SKA	• Compare performance to acceptable standard	• Provide feedback to learner
Transfer	Generalization	Allow for generalization of recall cues	• Provide collaborative learning exercises (team problem solving) • Provide alternative contexts in which SKA can be used	• Illustrate how new SKA might be used in new situation • Have learners generate new ways to use SKA
Valuing	Personalizing	Reinforce meaningfulness of new SKA	• Utilize SKA as context for new learning • Apply SKA in authentic activities	• Reinforce behavior by making it relevant to work or another new SKA to be learned

Source: AFMAN 36-2234, Instructional System Development[11]

and precise, and an elaborate method for simultaneous learning begins. An expert begins to emerge.

Fishburne et al.[12] noted that integration is such a complex process that training efficiency, if not effectiveness, could surely be improved if conditions fostering integration are identified and exploited during instruction. Useful terms in discussing integration are schemas, enterprise theory, and metaskills.

Schemas

Information is integrated into existing contexts or related knowledge to be remembered and recalled. This ordering in memory is thought of as schemas or elements representing a large set of meaningful information pertaining to a general concept. The concept may be of an object, such as a jet aircraft, a building, or a tree. Or it may be an event, such as preflight check or lightning strike. Regardless of type, schemas contain information on certain well understood features of the object or event. These features, called slots, are filled in by the learner when encountering new information that relates to the schema. Schemas are acquired through experience and may be the greatest benefit of apprenticeships. The application of schemas in instructional design is that a knowledge or skill should be learned and practiced in context of the "big picture" or broader, more encompassing concept, such as teaching the relationship between the electrical system and starting the engine.

Fishburne et al.[12] described a possible schema structure which would show patterns, relationships, situations and circumstances, similar schemas, approach, attention required, subgoals or checkpoints, rules, context, and dynamic patterns (timing, coordination, variation, cause-effect factors.) Again, this illustrates the complexity of learning. The formulation of schemas takes place within an enterprise.

Enterprise Theory

Gagné and Merrill[7] proposed a method to identify learning goals that require an integration of multiple objectives. They proposed that such an integration of multiple objectives be conceived in terms of the pursuit of a comprehensive purpose in which the learner is engaged, called **enterprise**. An enterprise is a purposeful, planned activity that may depend for its execution on some combination of verbal information, intellectual skills, and cognitive strategies, all related by their involvement in the common goal. A task for the instructional designer is to identify the goal of a targeted enterprise along with its component skills, knowledge, and attitudes, and then to design instruction that enable the student to acquire the capability of achieving this integrated outcome.

Accomplishment of schemas is an enterprise. The student works to a goal, and wants to achieve. Relationships between enterprises begin in part-task training, with normal procedures, simple activities, proficiency advancement, establishing relationships of part-to-whole. Building learning-on-learning, enterprises become more complex, variations such as emergency procedures are introduced, coordination among other players is practiced. Over time, broad expertise is developed as experience and confidence increase.

Relationships between enterprises begin in part-task training, starting first with normal procedures. The initial enterprises are simplified goals with proficiency required before proceeding to the next enterprise. Relationships are established part-to-whole and practice begins in integrating related enterprises. Through this process, learning is built upon learning and enterprises become more complex. Variations are introduced such as emergency procedures and interactions required with other players. The resulting complex enterprise performed proficiently under varying conditions must be organized in a manner that can be described in terms of hierarchical relationships. This is a macro approach to understanding enterprise components and the strategies for accomplishing them.

A good approach to understanding what must happen during successful development is to focus on the differences between the schemas in novices and those of experts. For example, in aircrew training there is a point called acquisition of initial proficiency where the new pilot can accomplish all the checklist items and perform the maneuvers correctly. There the integrated performance is that of the novice. The instructor recognizes that the right schemas for safe flight are formed, but the novice must continue to work other enterprises in order to gain "age and experience." The pilot continues from initial qualification to mission or combat qualification. The schemas are now

expanded to meet the demands of the "real world." The squadron commander knows that this is not enough. Exercises such as Red Flag, combat training in simulation, in-flight practice in multiple situation, each build on the schemas. In time the expert emerges, having achieved a repertoire of metaskills.

Figure 1 shows the progression from simple individual objectives to the more complex end goal or terminal objective. The integration of multiple objectives, or enterprises, is developed along the simple to complex continuum. The highest plateau of this continuum is the metaskill. Fishburne et al.[12] described the metaskill as the ability to adapt the specific skills it affects to the requirements of a situation. It is the essence of skill robustness. The more generalized the metaskill, the greater the variations among situations that can be accommodated; the more the metaskill is simultaneously discriminative, the more likely that given adaptations will be appropriate and precise. Metaskills thus serve as a high-level transfer system. The performer draws upon past experience to define and structure situation requirements and adapt skills accordingly. Fully developed, they are elaborate systems for simultaneous learning. In other words, the expert has metaskills.

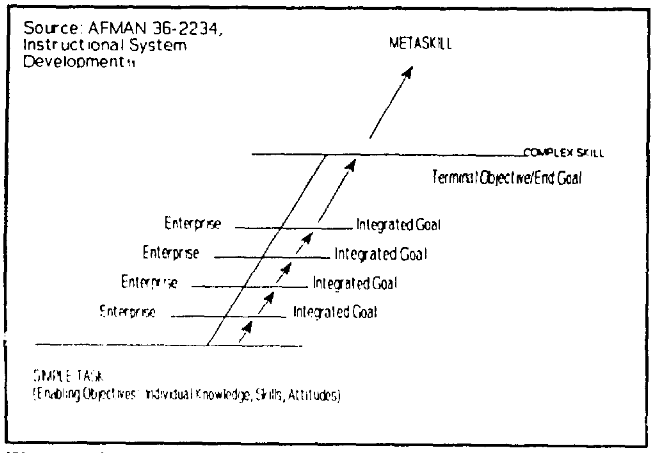


Figure 1. Progression from Simple Task to Metaskill

Metaskill

Spears[13] described a metaskill as the complex skill of adapting, monitoring, and correcting the use of individual skills in complex performances that integrate all learning processes. The person with a metaskill can deal with the novel situation successfully.

Media Selection for Integrated Activities

Gibbons[14] describes the method of media selection for integrated activities. Table 2 summarizes the approaches to consider when selecting media for integrated learning activities. Examples of learning activities for each approach are provided with possible media for supporting those activities. The instructional designer should consider the learning activity for which instruction is being prepared and select the appropriate media to achieve the integrated goal, end goal, or terminal objective.

Assessment of Integrated Activities

The method for assessing integrated activities is in context of the activity or event, such as low visibility takeoff or riding a bicycle in traffic. One approach is to use the degree of instructor involvement with the student. The 7-point scale starts with the student observing and the instructor demonstrating (1.0) and goes in half-point increments to the student performing beyond mere proficiency with no instructor intervention (4.0). Another approach is to use the degree of performance observed. This 4-point scale starts with performance that indicates lack of ability and knowledge (1) and goes to performance with unusually high degree of ability (4). Whatever the approach, a rating from the scale is given for each event accomplished during the training session. The form for recording the responses has the rating scale defined across the top of the form and the flight events listed by phase of flight down the side. There is a block next to each event for the rating and a space to the right for any notes the instructor feels important to jot down. This method has been used successfully for grading student progress in both simulation and inflight aircrew training.

Data from student ratings is a good measure for training effectiveness evaluation. Experience in using this data has shown that data should be summarized only by event. There is important understanding between events that is masked if an attempt is made to average ratings across events. The training decisions are also made event by event, such as when the benefit of training an event in simulation has been achieved and the remainder of training needs to be in the aircraft.

Table 2. Approaches for Media Selection for Integrated Learning Activities

Approach	Example of Activity	Example of Media
Provide alternate media for presentation and practice	Function of Parts	CBT
	Procedures	PTT or Simulator
Provide multiple media for the same task	Emergency Procedures	Classroom, CBT, Simulator
Provide intermediate practice exercises	Air Refueling	PTT, Simulator, Aircraft
Provide repeated, spaced practice	Landing an Aircraft	Simulator, Aircraft

Source: AFMAN 36-2234, Instructional System Development[11]

SUMMARY

The human process of learning is complex and is integrated in real life. The classification schemes and learning theories help us to understand the complex process, but the application to instructional design requires an integrated approach. We can decompose activities into skills, knowledge, and attitudes, and we can list tasks and missions. We can derive corresponding objectives for achievement of each task and define terminal objectives. However, we must not overlook the re-integration of these elements back into the "whole." We build hierarchical relationships, form enterprises of multiple objectives and establish instructional strategies. We always keep in context the "big picture," keeping the student in tune with "how it all fits." We plan beyond initial skill acquisition, preparing for the full life cycle of education and training requirements. We design for metaskill development over the long term. We recognize that the complexities of learning coupled with individual differences cause us to be flexible, willing to adjust to the needs of the moment. We admit today we still do not have all the answers for tomorrow. We will continue to seek a better way.

REFERENCES

1. Bloom, B. S. *Taxonomy of Educational Objectives*. New York: McKay, 1956.
2. Gagné, R. M. *The Conditions of Learning*. (4th Ed.) New York: Holt, Rinehart and Winston, 1985.
3. Merrill, D. Component Display Theory. in C. M. Reigeluth (Ed.), *Instructional Design Theories and Models: An Overview of Their Current Status*, Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1983, 279-333.
4. Reigeluth, C. M. *Instructional Design Theories and Models: An Overview of Their Current Status*. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1989.
5. Skinner, B. F. *Technology of Teaching*. New York: Appleton, 1968.
6. Mager, R. F. *Preparing Objectives for Instruction* (2nd Ed.). Belmont, California: Fearon Publishers, 1962.
7. Gagné R. M. and Merrill, D. Integrative Goals for Instructional Design. *Educational Technologies*, 38(1), 1990, 22-30.
8. Keller, J. M. The Systematic Process of Motivational Design. *Performance and Instruction*, 26(9), 1987, 1-8.
9. Gagné, R. M. Discussion with authors, 1992.
10. Tennyson, R. D. Educational Learning Theory for Instructional Design. *Educational Technology*, 32, 1992, 36-41.
11. AFMAN 36-2234, Instructional System Development, U. S. Air Force, 1993.
12. Fishburne, R. P., Williams, K. R., Chatt, J.A and Spears, W. D. *Design Specification Development for the C-130 Model Aircrew Training System: Phase I Report*, Williams AFB, Arizona: Air Force Human Resources Laboratory, AFHRL-TR-86-44, 1987.
13. Spears, W. *Processes of Skill Performance: A Foundation for the Design and Use of Training Equipment*. (NAVTRAEQ-VIPCEN 78-C-0113-4). Orlando, Florida: Naval Training Equipment Center, 1983.
14. Gibbons, A. S. Adding an Expert to the Team: The Expert Flightplan Critic. *Interactive Learning, International*. Volume 6, No. 2 (63-72). London, United Kingdom: John Wiley & Sons, Ltd., 1993.

The Use of Computer-Based Videogames in Knowledge Acquisition and Retention

Katrina E. Ricci
Naval Training Systems Center
Orlando, Florida

ABSTRACT

The cost associated with the use of desktop computer-based games as an instructional technique is minimal. However, the potential of computer-based videogaming as an effective training approach has not been determined. There are several reasons for expecting effective knowledge acquisition and retention using computer-based videogaming. First, videogames may combine principles of computer-assisted instruction, such as the contiguity between the stimulus and response, knowledge of results, and practice (Driskell and Dwyer, 1984). Second, properties of computer-based videogaming such as active participation, competition, and challenge against uncertain outcomes, have been associated with increased motivation to participate in videogaming (Shrestha, 1991; Malone, 1984). Finally, research has found that testing during training, one aspect of videogaming, will increase retention scores (Hogan & Kintsch, 1971; Hagman and Rose, 1983).

Research conducted at the Naval Training Systems Center investigated the acquisition and retention of basic knowledge with subject matter presented either in paper-based prose form (TEXT), paper-based question and answer (TEST) form, or using videogaming techniques (GAME). These conditions were selected to investigate potential benefits of videogaming over traditional paper and pencil media and to identify the extent to which benefits obtained from videogaming could be due to testing during training. Results showed subjects assigned to the GAME condition scored significantly higher on a retention test as compared to pretest performance. Subjects assigned to the TEST and TEXT conditions showed no difference in performance from pretest to retention test. Additionally, subjects assigned to the GAME condition rated the training they received as more enjoyable and more effective than those assigned to the other two conditions. Results are discussed in terms of the effectiveness of computer-based games and applications for military training.

ABOUT THE AUTHOR

Katrina Ricci is a Research Psychologist with the Naval Training Systems Center, Orlando, FL. She has a Masters of Science in Industrial/Organizational Psychology from the University of Central Florida and is currently working toward a Ph.D. in Human Factors Psychology. Ms. Ricci serves as Principle Investigator for the Serious Pursuit/GameShell Cooperative Research and Development Agreement, Principle Investigator for an Exploratory Development Skill Retention Project, and Assistant Project Manager for the Electronic Warfare Continuum Assessment Program for Naval Aviation.

The Use of Computer-Based Videogames in Knowledge Acquisition and Retention

Katrina E. Ricci
Naval Training Systems Center
Orlando, Florida

INTRODUCTION

Videogaming has been defined as a rule-governed, goal-focused, microcomputer driven activity incorporating principles of gaming and computer assisted instruction (CAI). A PC-based videogame training system is normally employed as a stand-alone device requiring minimal instructions and monitoring, and consequently, a reduced amount of instructor time (Driskell & Dwyer, 1984). Therefore, the cost associated with the use of desktop computer-based games as an instructional technique is minimal.

The potential of computer-based videogaming as an effective training approach has not been determined. There are several reasons to expect effective knowledge acquisition and retention using computer-based videogaming. First, videogames may combine principles of computer-assisted instruction, such as the contiguity between the stimulus and response, knowledge of results, and practice (Driskell and Dwyer, 1984). Second, properties of computer-based videogaming such as active participation, competition, and challenge against uncertain outcomes, have been associated with increased motivation to participate in videogaming (Shrestha 1991; Malone, 1984). Finally, research has found that testing during training, one aspect of videogaming, will increase retention scores (Hogan & Kintch, 1971; Hagman and Rose, 1983).

Background

In 1989, the Naval Training Systems Center (NAVTRASYSCEN) developed a computer-based version of a board game designed to teach sailors and marines about the Soviet Union. The game, Serious Pursuit, presented questions in five categories of information: Soviet Navy, other Soviet Services, Soviet people, Soviet Geography, and Soviet History. Serious Pursuit was distributed to over a thousand fleet units and was very well received.

Since that time, two other variations of the Serious Pursuit game were developed: Chemistry Pursuit and Contract Trivia. A number of other subject matter games were requested, as many military jobs require a pre-existing knowledge base in order to carry out tasks. However, due to the extensive programming time involved, the process of creating additional Pursuit type games was not optimal.

In 1991, NAVTRASYSCEN entered a Cooperative Research and Development Agreement (CRADA) with Dynamics Research Corporation to develop a software program, or shell, to house question and answer data bases. Once entered into the shell, these questions can be selected for inclusion in a videogame. After the game is played, the software collects performance data and generates student reports. The program developed, GameShell, enables the creation of computer-based games without additional software programming. GameShell is now available on the commercial market. In accordance with the CRADA, the same version of this program, QuizShell, is available for Department of Defense use.

QuizShell

Questions, associated answers, and feedback material entered into QuizShell files can be selected to generate a computer-based game. When the game is played, a student is presented with three boxes that operate in slot machine fashion. Each box randomly selects and presents a question category that has an associated value of 50 points. Students choose a question category from those

available in the boxes. A question from that category is then displayed and the student is given three minutes to answer. If answered correctly, players receive points toward their score. Once players have answered a predetermined number of questions per category and have earned a set number of points, the player "wins".

Computer-based educational games generally will fall into two categories: simulation games and videogames. Simulation games model a process or mechanism relating input changes to outcomes in a simplified reality that may not have a definite end point. They often depend on the learner reaching conclusions through exploration of input changes on outcomes. Videogames, on the other hand, are competitive interactions bound by rules to achieve specified goals that are dependent on skill and often involve chance and imaginary settings (Randel, Morris, Wetzel, & Whitehill, 1992). While QuizShell differs slightly from traditional videogames in that the game is predominantly text, it does provide a starting point to determine the instructional and motivational benefits of videogaming.

Effectiveness of Games as an Instructional Technique

From an instructional point of view, videogames offer two important aspects of training: practice and feedback. Videogame practice is not dependent on instructor time and can continue as long as computer time is available. Immediate feedback on performance is also provided. The QuizShell game offers feedback after each question is answered. At a minimum, the player is told his response is correct or incorrect, and is provided the correct answer. Depending on the developer, extended feedback on the correct answer may also be given.

In a review of the effectiveness of games for educational purposes, Randel et al., (1992) reviewed 67 investigations covering a period of 28 years. Of the research reviewed, 38 found no difference between games and conventional classroom instruction, 22 favored the use of games, five favored games but used questionable control groups, and only three favored conventional instruction. Ten of the 14 efforts measuring retention reported significant effects favoring simulation and gaming for retention, whereas the remaining four found no difference in retention between the games and conventional instruction. Twelve of 14 investigations showed student interest in game conditions higher than in traditional classroom approaches. Randel et al. concluded that subject matter areas are more likely to show beneficial effects for gaming when very specific content can be targeted.

Testing During Training

In experiments manipulating presentation and test conditions, information to be learned is presented by the experimenter. Under presentation conditions, information is presented to subjects for study. Under test conditions, the information is removed and recall is tested. Research has found that testing during training, one common aspect of videogaming, will increase retention scores as compared to presentation alone. This has been found in motor skill retention research (Hagman, 1980) and verbal list learning research (Hogan and Kintsch, 1971). The benefits of testing on retention are attributed to superior learning, or encoding, of the task during test trials. Testing provides a feedback mechanism that allows the learner to clarify what has been learned and to identify what remains to be learned. Testing may also facilitate later recall of learned information.

Motivation

Several properties of computer-based videogaming have been associated with increased motivation to participate in videogaming. These properties include active participation, competition, and challenge against uncertain outcomes. Shrestha (1991) found subjects assigned to a videogame condition were willing to spend longer periods during training as opposed to those in a non-game condition. Seymour, Main, Randel, & Morris (1992) found motivation to study had a significant positive correlation with success in difficult tests in school.

RESEARCH OBJECTIVES

The research described here investigates the acquisition and retention of basic knowledge with subject matter presented either in paper-based prose form (TEXT), paper-based question and answer (TEST) form, or using videogaming techniques (GAME). These conditions were selected to identify potential benefits of videogaming over traditional paper and pencil media and to identify the extent to which benefits obtained from videogaming could be due to testing during training.

METHOD

Subjects

Sixty students, 56 male and 4 female, assigned to the Electronics Technical 'A' School, Naval Training Center, Orlando served as subjects for this experiment. The median age of the participants was 20 years. Prior to testing, subjects were informed as to the general nature of the experiment, and were required to read and sign an informed consent form. Subjects were randomly assigned to one of three training groups, twenty subjects per group: GAME, TEST, and TEXT.

Apparatus

Subjects assigned to the GAME condition played the QuizShell game on a Zenith 248 computer. The game was created using the software developed under a CRADA between the NAVTRASYSCEN and Dynamics Research Corporation and contained questions in five categories of chemical, biological, and radiological defense (CBRD). The training materials used for the TEXT condition was a Chemical and Biological Defense (CBD) Common Skills Pocket Handbook. The handbook was originally developed by NAVTRASYSCEN and later updated by the Institute for Simulation and Training under the support of the U.S. Army Dugway Proving Ground, Dugway, Utah. The materials for the TEST condition were generated using the questions and answers from the QuizShell CBRD game.

Prior to the training task, subjects completed a brief demographic questionnaire used to determine subject age, gender, length of time in service, and time since recruit training. Following the training task, subjects completed a brief subjective opinion questionnaire on the training task. The opinion questionnaire contained eight questions to be answered using a five point Likert scale (1 = strongly disagree to 5 = strongly agree).

Training Tasks

__Game__. Subjects assigned to the GAME condition played the CBRD game. The game was constructed of 88 questions across five question categories: Agents, Decontamination, Detection, Protective Gear and Radiology. Questions were presented after subjects picked a category from the main game display. After selecting a category, the question was displayed, the subject selected an answer, and feedback (correct, incorrect) was given. The correct answer was always displayed after the subject's response was made. The subject was then brought back to the main game display to select the next topic. Points were scored by answering questions correctly; more points were available if the topic selected was available in more than one box on the main game display. Subjects were instructed to play the game until they had answered four questions correctly in each category and had scored a total of 2500 points.

__Test__. Subjects assigned to the TEST condition reviewed the same 88 questions available in the QuizShell CBRD game, but presented on paper. Immediately after each question, the correct answer and any other supporting feedback available the QuizShell game were presented.

__Text__. Subjects assigned to the TEXT condition reviewed the 63 page Chemical and Biological Defense Pocket Handbook.

All information needed to answer post tests were available in all three training tasks. Irrelevant information was present in all three conditions; however, the amount of irrelevant information was comparable over all three conditions.

Procedure

Prior to participating in this experiment, each subject read and signed an informed consent form. Subjects then completed the demographic questionnaire and a 20-item, multiple choice, pre-test.

During the 45 minute training period, subjects assigned to the GAME condition played the CBRD QuizShell game developed for this experiment. Subjects assigned to the TEST condition had 45 minutes to study the same questions and answers available in the CBRD game, but in paper form. Subjects assigned to the TEXT condition had 45 minutes to study the CBR Pocket Guide.

Following training, all subjects completed a brief opinion questionnaire and a 20 item, multiple choice, post-test.

Four weeks after the initial training session, all but two subjects assigned to the TEXT group returned to take a 20-item, multiple choice, retention test. All test questions in the post and retention tests were derived from information available in all three conditions. All three tests (pre, post, and retention) were of parallel form and did not overlap in question content.

RESULTS

The structure for this analysis was a 3 X 3 repeated measures design with one within-subjects factor and one between-subjects factor. The within-subjects factor, test phase, had three levels: pretest, post-test, and retention test. The between-subjects factor, group, had three levels: GAME, TEST, and TEXT. Pretest, post-test, and retention test score means by group are given in Table 1.

Test Phase

The within-subject analysis showed significant differences for test phase, $F(2,110)$, $p<.001$. Planned comparisons revealed all three groups performed significantly better on the post-test as compared to the pretest: $t(19)=-10.03$, $p<.001$; $t(19)=-5.22$, $p<.001$; and $t(19)=-7.59$, $p<.001$ for the GAME, TEXT, and TEST conditions, respectively.

Conversely, all three groups performed significantly worse at the retention test compared to post-test performance: $t(19)=7.93$, $p<.001$; $t(17)=3.62$, $p<.002$; and $t(19)=11.81$, $p<.001$ for the GAME, TEXT, and TEST conditions, respectively.

Table 1. Means by group and testing phase for test percentage correct[1]

	Testing Phase		
Group	Pre	Post	Retention
GAME	52.0 (10.7)	82.0 (8.3)	59.3 (10.8)
TEST	52.5 (13.9)	86.0 (15.9)	56.0 (12.3)
TEXT	48.3 (6.7)	65.0 (11.8)	49.7 (13.8)

[1]Means of test scores are in normal typeface; standard deviations are in parentheses.

While there were no significant differences between the pretest scores and retention test scores for the TEXT and TEST conditions, subjects assigned to the GAME condition performed significantly better on the retention test than on the pre-test, $t(19)=-2.49$, $p<.05$.

Group

The between-subjects analysis showed significant differences for group ($F(2,55)$, $p<.001$. Planned comparisons among means were performed using the Student Newman-Keuls method. There was no significant difference found between the GAME, TEST, and TEXT conditions on pre-test and retention-test performance. However, further analysis showed the TEST and GAME groups performed significantly better than the TEXT group on the post-test ($p<.05$).

Subjective Opinion Scores

Separate analyses were conducted to analyze the opinion questionnaires. The questions were answered on a 5 point Likert scale: 1 =

strongly disagree to 5 = strongly agree. The five questions relating to the training conditions were as follows:

1) This form of study was enjoyable.
2) I learned a lot about CBRD during today's training session.
3) I feel confident I will remember what I learned today.
4) I would prefer this form of study in my other Navy courses.
5) This program wasted my time.

Mean rating scores from the three conditions fell consistently in the same pattern (Table 2). Five separate ANOVAs were conducted, one for each of the five questions associated with training method. In the presence of significant main effects, post hoc comparisons among means were performed using the Student Newman-Keuls method.

Table 2. Means of subjective opinion scores by question and group

	GROUP		
Question	GAME	TEST	TEXT
1	3.35	2.50	2.00
2	3.60	3.20	2.78
3	3.30	3.05	2.11
4	3.35	2.60	1.89
5	2.10	2.60	3.22

Questions 1, 3, 4, and 5 all showed significant main effects for group, $F(2,55)$, $p<.002$. Further analyses showed subjects assigned to the GAME condition gave significantly higher ratings than those assigned to the TEST and TEXT conditions for Question 1 ($p<.05$); subjects assigned to the TEXT condition gave significantly lower ratings than those assigned to the TEST and GAME condition for Question 3 ($p<.05$); subjects assigned to the GAME condition gave significantly higher ratings than those assigned to the TEXT condition for question 4 ($p<.05$); and subjects assigned to the GAME condition gave significantly lower ratings than subjects assigned to the TEXT condition for Question 5 ($p<.05$).

DISCUSSION

It was hypothesized that the GAME condition would result in superior acquisition and retention scores as compared to the TEST and TEXT conditions. While all subjects showed an increase in performance from pretest to post-test, the GAME and TEST subjects performed significantly better at the post-test than the TEXT subjects. Only subjects assigned to the GAME condition achieved significantly higher retention scores as compared to their pretest scores.

These results suggest that the methodology of information presentation utilized in the TEST and GAME conditions allowed learners to focus directly on the concepts that were critical. Further, those subjects assigned to the TEST and GAME conditions had the opportunity of active participation in the training conditions with the ability to test their knowledge during training.

The superior retention of the GAME condition is probably the result of more focused attention to the training materials attributable to interest in and motivation to the type of practice. As expected, subjects assigned to the GAME condition rated the training they received as more enjoyable and more effective than those assigned to the other two conditions.

CONCLUSIONS

Today's military personnel are required to be knowledgeable in a number of areas. Training is necessary not only for complex performance skills, but also for factual information. Traditional classroom approaches for teaching knowledge are not always enthusiastically received by young service members who have grown up in a era of computers and computer gaming.

Results described here show videogaming as a potentially effective method for knowledge acquisition and retention. Basic rules of

learning (i.e., practice and feedback) applied in a videogaming setting may increase student motivation to study, and therefore. ultimately increase their knowledge store.

A number of applications of videogaming can be found for military training. Introduction to novel material may be facilitated by contact through videogaming. Due to the fact that human memory is by nature perishable, videogaming can also be utilized in determining the need for and conduction of refresher training. Videogaming can also be applied to situations where expensive simulations or training systems are only available on a person by person basis, where the trainee would otherwise be waiting for the opportunity to train. Finally, videogames made available to recreation rooms or break areas can foster healthy competition and indirectly serve as training mechanisms.

REFERENCES

Driskell, J. E. & Dwyer, D. J. (1984). Microcomputer videogame based training. *Educational Technology,* 11-16.

Hagman, J. D. (1980). *Effects of presentation- and test-trial training on motor acquisition and retention* (TR 431). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Hagman, J. D. & Rose, A. M. (1983). Retention of military tasks: A review. *Human Factors, 25(2),* 199-213.

Hogan, R. M. & Kintsch, W. (1971). Differential effects of study and test trials on long-term recognition and recall. *Journal of Verbal Learning and Verbal Behavior, 10,* 562-567.

Malone, T. W. (1984). Heuristics for designing enjoyable user interfaces: Lessons from computer games. In J. C. Thomas and M. L. Scheider (Eds.), *Human Factors in Computer Systems.* Norwood, NJ: Ablex Publishing Company.

Randel, J. M., Morris, B. A., Wetzel, C. D., & Whitehill, B. V. (1992). The effectiveness of games for educational purposes: A review of recent research. *Simulation & Gaming, 23(3),* 261-276.

Seymour, G. E., Main, R. E., Randel, J. M., & Morris, B. A. (1992). *Study factors and their impact on military school performance measures* (NPRDC TR-92-10). San Diego, CA: Naval Personnel Research and Development Center.

Shrestha, L. B. (1987). *Computer based training: The effects of game characteristics on motivation and learning.* Unpublished master's thesis, University of Central Florida, Orlando, FL.

BYTES vs. BULLETS
CREW-SERVED WEAPONS TRAINING AND SIMULATION

Brian K. Wilhoite
United States Marine Corps
School of Infantry, Marine Corps Base
Camp Lejeune, North Carolina

ABSTRACT

Can crew-served weapons training in the military be augmented by simulation? - YES Simulation is going to play an ever increasing role in the nation's ability to maintain combat effectiveness in the armed services. While cutting personnel and training dollars, the services are attempting to get leaner and more efficient. Simulation is proving to be beneficial for crew-served weapons training. Tests and responses to recent crew-served weapons training conducted with the improved Indoor Simulated Marksmanship Trainer (ISMT) at the School of Infantry, Camp Lejeune, North Carolina have proven that simulation is an effective for augmenting the training of crew-served weapons teams.

ABOUT THE AUTHOR

Brian Wilhoite recently left duties as the Training Systems Support and Plans officer for the School of Infantry (East). He was the simulation project officer in charge of simulation testing and evaluation of the 3rd generation ISMT at the School of Infantry, Marine Corps Base, Camp Lejeune, North Carolina. Since his commissioning eight years ago as a Marine Officer, he has either been in command of or directly involved with the training of Marines. Brian is currently attending the Marine Corps' Amphibious Warfare School in Quantico, Virginia.

BYTES vs. BULLETS
CREW-SERVED WEAPONS SIMULATION BASED TRAINING

Brian K. Wilhoite
United States Marine Corps
School of Infantry, Marine Corps Base
Camp Lejeune, North Carolina

INTRODUCTION

As a result of reductions in the armed services, simulation based training will play an increasing role in maintaining the nation's combat effectiveness. Simulation has clearly been identified as a cost effective method of maintaining specific, mission-essential skills while reducing training resources. For years simulators have effectively augmented pilot, tank, and other high-cost training. Advanced technology now provides economical simulation to less complicated environments.

Studies on marksmanship simulators are abundant. Numerous publications exist pertaining to the replacement and augmentation of M-16 live fire training with simulation. But this training has been focused on individual soldier training, not crew or collective training.

Can crew-served weapons training in the military be augmented by simulation? Initial test indications say yes. The Marine Corps is currently studying the effects of simulation for crew-served weapons training using the Indoor Simulated Marksmanship Trainer (ISMT). The ISMT tested and discussed in this paper has seven key components and is described in figure 1.

Ground Weapons Simulation: Background

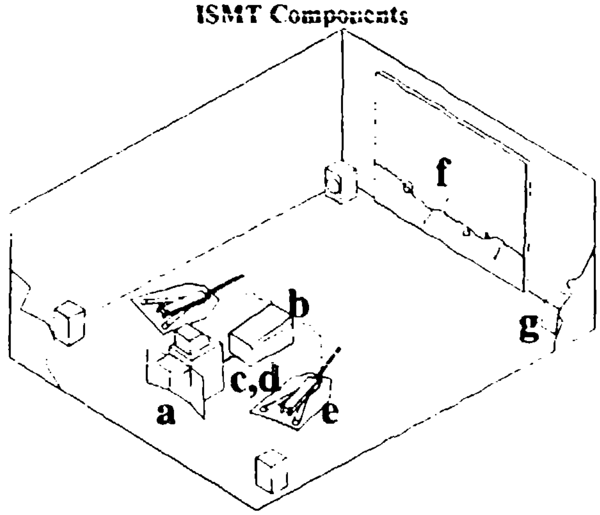
The U.S. Army was the first to test a simulator's ability to assist in teaching collective skills. This testing was conducted with Oregon National Guard infantry and support squads using the Squad Engagement Training System (SETS), (Eisley, 1990). SETS was used to test both marksmanship and command and control skills. Marksmanship results were provided by the simulator and the command and control evaluation of the squad leader was conducted by subjective evaluation. The test produced results that indicate that simulation could be used as a tool to teach collective skills; however, test results were not conclusive because of flawed testing procedures (TRADOC TEE Memo, 1990).

Until recently, the Marine Corps has concentrated only on the use of simulation for individual marksmanship. It fielded marksmanship simulators designed specifically for remediation of shooters having difficulty qualifying with the M16 service rifle (USAF,1992, CNA,1987). These second generation marksmanship trainers have recently been upgraded. The new, third generation provides for crew-served weapons training.

INDOOR SIMULATED MARKSMANSHIP TRAINER(ISMT)

Figure 1

ISMT Components



The ISMT is made up of the following components:

(a) Central Processing Unit
(b) Color Video Projector
(c) Video Disc Projector
(d) Laser Disk Media
(e) Demilitarized Weapons
(f) Large Screen Display
(g) Surround sound audio

The third generation system is designated the Indoor Simulated Marksmanship Trainer (ISMT). The ISMT provides a quantum leap in small caliber, ground weapons training capabilities. For the first time, real consideration can be given to replacing or augmenting various types of live fire training

BYTES VS. BULLETS

The School of Infantry is currently evaluating the simulation capabilities of the ISMT. This evaluation is designed to measure the systems ability to maintain or enhance our ability to train entry level Marines. The impact of simulation training is being compared to the historical results obtained using traditional instructional methods.

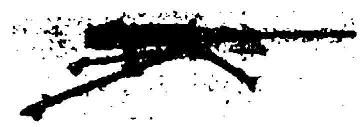
In February 1993, the Marine Corps fielded five test ISMT units. The ISMT provides simulation training for every weapon organic to a Marine Corps infantry battalion. In particular it is the first multi-dimensional crew-served weapons simulator. The system was fielded with the following weapons types:

a) M16 Service Rifle
b) M9 Service Pistol
c) M203 Grenade Launcher
d) M249 Light Machinegun
e) M60 Medium Machinegun
f) M2 Heavy Machinegun
g) MK19 Automatic Grenade Launcher
h) AT-4 Light Anti-Armor Assault Weapon

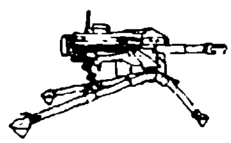
The School of Infantry, Camp Lejeune, North Carolina received two of the five systems, with the charter to test the simulators' capability to train crew-served weapons students.

The School of Infantry has successfully integrated the use of the ISMT into 0331 machinegunnery military occupational specialty (MOS) training. Students in the 0331 program of instruction (POI) are entry level Marines receiving their initial MOS qualification. Mastery of three different machineguns is required for successful completion of this course of instruction. These machineguns include the M60 medium machinegun, M2 heavy machinegun, MK-19 automatic grenade launcher.

M60 Medium Machinegun



M2 Heavy Machinegun



MK-19 Automatic Grenade Launcher

Based on the ISMT's advertised ability to simulate all of the Marine Corps' family of machineguns, we selected the 0331 POI as our target evaluation group. Machinegun training is conducive to simulation for several reasons.

Target Engagement - The successful employment of the weapons systems is measured by the gun teams' coordinated effort in successfully engaging the wide range of targetry afforded by simulation. Successful target engagement is a measure of several combined skills:

a) The assistant gunners verbal commands to the gunner such as target description, range, and supplementary corrections after initial engagement.

b) The gunners ability to manipulate the weapon from directions given by the assistant gunner.

c) The ability of the team to keep the weapon system functioning and in action.

Physiological Advantages - Many skills required to be a good machinegunner are physiological and learned

through repetition. Simulation offers a cost effective means to conduct repetitive training.

Feedback - Immediate feedback from the simulator would encourage timely corrections to marksmanship tasks being taught. This feedback drastically improves the learning curve of both basic and intermediate marksman.

Increased Trigger Time - Crew-served weapon teams require more firing time to become an effective fighting force than do Marines learning individual marksmanship skills.

Reduced Ammunition Costs - The cost of crew-served weapons ammunition is high relative to other small caliber munitions.

Affordable - The apparent cost effectiveness of state of the art simulation (e.g. the ability to replace select quantities of live ordnance and still meet training standards).

Reduced Training Costs - The apparent cost savings offered over traditional field training.

a) Logistical costs of moving training to a field environment.

b) The time and equipment required to prepare and execute field training.

TEST PLAN

Class 12-93, Company "A", of the Infantry Training Battalion was selected as our first test class. Their training was conducted from March to April of 1993. Class 12-93 had 26 machinegunnery students that underwent instruction. This was an average class size for the 0331 MOS (see table 1).

This class was divided in half using the last digit of each student social security number to distinguish the test group from the control group. Those Marines whose social security numbers that ended in odd numbers were placed in the simulation group. Those ending in even numbers made up the control group. It turned out that 13 Marines were in each group

Test Variables

The control group followed the normal program of instruction. The test group was provided select periods of simulation that replaced the traditional gun drills practiced by the control group. The simulation sequences fired by the test group were carefully selected to emphasis the same training tasks taught and practiced during gun drill.

Gun drill is gunnery training that requires student to conduct specific mechanical manipulations to the machinegun after receiving commands from their assistant gunner.

Our test group conducted simulated target engagement training. Successful target engagement is the evaluation process of tasks taught during gun drill. The test group received the added benefit of immediate feedback during their target engagement training. The ISMT provides an outstanding tool for individual feedback for <u>every round fired</u>. This provides diagnostic review of each machinegun teams target engagement abilities.

Table 1 shows a comparison from all of the classes for fiscal year 1993. The results of class 12-93 are in Table 2.

Table 1

FY 1993 Machinegunnery Qualification Scores

Qualification Scores by Weapons System	C 93-1	C 93-2	C 93-3	C 93-4	C 93-6	C 93-7	C 93-8	C 93-9	C 93-10	FY-93 AVERAGES
M60	81	88	97	80	77	82	82	83	72	**82**
M2	73	90	83	85	79	86	76	74	85	**81**
MK19	93	100	79	90	96	89	96	85	95	**91**
									Composite Average	**85**
# Students per class	16	20	16	37	31	26	15	19	32	**24**

*There was no machinegunners in class 5-92

Results

Several operational problems kept this from being a purely scientific test. This class was the first to use simulation integrated into the POI. Secondly, the Mk19 Automatic Grenade Launcher simulators were not fully operational for this class. Third, for morale purposes, all machinegunners in the class got a basic ten minute demonstration exercise on the M60 machinegun simulator.

One of the research weaknesses noted in other simulation tests is that the test group receives simulation training in addition to traditional training. The test group and the control group are then scored on a live fire test. The test group usually out shoots the control group. The question that begs asking is; "How much of the increase was due to the fact that the test group received additional training and how much is directly attributable to the quality of the simulation?"

Although our test group was out scored on two of the three events, their overall average was higher than the control group. These numbers are only part of the picture (see table 2). The MK-19 simulator was not fully functional for the test so those scores have limited impact on the experiment. Secondly, the M2 qualification is the final qualification fired and each student in the test group received three periods of instruction using the simulator prior to this qualification shoot. The overall outcome of the test group was a total 1% increase over the control group. More significantly though, it was 3% higher than the annual average.

The most significant result the test produced was the significant improvement of the test group over the annual average. This significant increase in score can be attributed to a very limited amount of simulation. Each Marine in the test group only received a total of 20 minutes of simulation time on the M60 and the M2 machineguns. The 0331 program of instruction is composed of 232 hours of lecture, demonstration, and application. The 20 minutes of simulation equates to only a .14% replacement of the POI. This .14% of simulation produced a 3% increase over the annual average.

Table 2

Machinegunnery Qualification Scores

Simulation Test Group vs. Simulation Control Group

Class 12-93	M60	M2	MK19	Averages
Test Group	85	84	96	88
Control Group	87	78	97	87
Annual Average	82	81	91	85

Cost Savings - Simulation used responsibly, will provide combat effective training in an increasingly austere fiscal environment. A cost analysis of simulation vs. live fire training provides an even clearer picture of the tangible advantages to simulation. Simulation provides moving targets and diagnostic feedback which is not available from most field training ranges. Other tangible savings, in addition to those shown in table 3, include:

a) Reduction in weapons usage & maintenance.
b) Reduction in equipment/personnel required to conduct live fire training:
   (1) Vehicles
   (2) Radios
   (3) Corpsman
   (4) Safety Personnel
c) Reduction in live fire range usage.
d) Reduces the requirement for specialty ranges if a virtual environment is conducive to the training required.

CONCLUSIONS

Crew-served weapons training can be augmented by simulation training. Table 3 gives a sample of what limited live fire augmentation to the School of Infantry (East) provides in ammunition savings. It is safe to estimate that equivalent savings could be expected from the School of Infantry (West). A total of $2.5M will purchase enough ISMT systems to meet the simulation requirements of both campuses of the School of Infantry.

With commensurate savings in other weapons systems such as the AT-4, M203, and M16, our experience shows that the School could pay for its simulation requirements within two fiscal years using conservative simulation quantities.

Table 3

Simulation Cost Analysis for the School of Infantry (E)

	# rounds per student	# rounds to be simulated	# of 0331 students per year	Total # of rounds fired per year	Total # of rounds to be simulation	% of proposed simulation	$ per round	**Annual Savings**
M60	3,600	2000	500	1,800,000	1,000,000	56%	$0.27	$270,000
M2	600	200	"	300,000	100,000	33%	$1.22	$122,000
Mk19	200	60	"	100,000	30,000	30%	$11.48	$344,400
							TOTAL	**$736,400**

Note(1): Information based on FY 92 ammunition costs

Note(2): Live fire training evoultions to be simulated include prequalification live fire evolutions that are used to train a Marine to preliminary standard

The plan:

a) Approximately 30% simulation of each weapon system.

b)The purchase of a minimum of six ISMT systems at $200K, for each School of Infantry campus.

We have continued to collect data from other student classes with similar initial results to those published herein. Appendix A provides a sample of responses collected from the machinegunners from class 12-93. Conclusions from this objective data show that simulation had a dynamic impact on the quality of instruction received by this class. Appendix B is a similar collection of responses from the 0311 Basic Infantryman course class 12-93. The riflemen of this course were also exposed to simulation as a group during selected portions of their POI.

As the simulation venture increases in speed some caution is required. Crew-served weapon training is being successfully conducted but certain eventualities relating to live fire training can not be replicated. Until simulation results are measured over time, it is prudent to measure the quantity of live fire training relegated to the virtual arena.

Through our ISMT simulation experience a clear methodology emerged. An ideal mixture of simulation vs. live fire can not be generated.

The following are lessons learned:

Individual Training Standards (ITS) -The ISMT is capable of training machinegunners to all ITS's pertaining to marksmanship published by the Marine Corps. These standards should be used, without modification, and without exception regardless of the type of training being conducted (e.g., simulation or traditional).

Task Mastery - All standards can be fired to mastery using simulation PRIOR to conducting live fire training.

Live Fire Preparation - Live fire training should be conducted to test skills mastered during simulation. Live firing becomes a confir ation process of simulation training.

Resolves Training Deficiencies - Simulation can provide off the shelf solutions to expensive training deficiencies. Examples:

a) Provides ability to engage targets at maximum effective range regardless of weapons system (e.g., M2 heavy machineguns historical live fire trange limitations).

b) Provides ability to engage moving targets at varying degrees of difficulty and obtain battle damage assessments.

c) Provides target arrays which give immediate scores. Additional benefits:

(1) Reduces reliance on undependable mechanical targets.

(2) Saves time taken to physically score targets for long range weapons.

(3) Provides immediate feedback for every round fired. This is especially important for target arrays normally located in impact areas where physical scoring is impossible.

Provides Remediation & Diagnostics - Provides an invaluable tool to identify individual and team weaknesses and allows for remediation of identified shortcomings without a reliance on available ammunition. No commander will be placed in a

dilemma of whom to train when ammunition shortages occur.

Due to planned budgetary reductions, the military must train efficiently and more economically. Simulation is a viable answer to cost effective training. Training ammunition makes up a significant percentage of all the services budgets. How much preparation for warfighting can we afford to do in a virtual environment? This question requires a collective, mature approach as cost effective training becomes a national security issue in these fiscally austere times.

Selective simulation provides an outstanding *augmentation* to current training practices. No one can responsibly advocate complete replacement of live fire training.

What must be kept in mind, above all else, is that warriors will still need to train in conditions as close as possible to the real thing. Warfare is not safe and realistic training must continue to carry with it some inherent dangers. The defenders of our nation will need to "smell the cordite" for years to come.

REFERENCES

1. Berg, R., Adedeji, A., and Steadman, G., "A Cost-and-Effectiveness Methodology for Evaluating the Use of Simulators in Marine Corps Live-Fire Training", Center for Naval Analysis, CRM 92-162, Feb. 1993.

2. Berg, R., Adedeji, A., and Steadman, G., "Simulation Offset to Live-Fire Training Study: Assessment of Marine Corps Live-Fire Training Support", Center for Naval Analysis, CIM 238, Jan. 1993

3. "Training Effectiveness Evaluation (TEE) of Squad Engagement Training System (SETS)," TRADOC Memo. ATSH-TDS-SM (350), ref: Trip Report, USAIS, ATSH-TD-S-D, 6 Oct 89, Subject: Squad Engagement Training System (SETS). Jul. 1990.

4. Eisley, M., Hagman, J., Ashworth, R., and Viner, M., "Training Effectiveness Evaluation of the Squad Engagement Training System (SETS)," AD-A226 406, U.S. Army Research Institute, Jun. 1990.

5. Firearms Simulation Test: Final Report, AF SPA/SPLT, Kirtland, AFB, NM, U.S. Air Force Security Police Agency, May 1992.

6. Gates, S., "Appropriate Mix of Live Fire and Simulated Fire During Training," Center for Naval Analysis, Research Memo. 87-116, Jul. 1987.

Appendix A

School of Infantry (E)
Infantry Training Battalion Class 12-93
Machinegunnery Simulation Questionnaire

	Outstanding	Good	Adequate	Needs Improvement	Poor	No Response
How well did the simulator meet the following areas?						
Safety	13	7	5		1	1
Loading	10	8	7	2		
Unloading	10	8	9			
Reloading	9	8	5	3		
Rapid Reloading	8	8	8	1		2
Cleaning	9	3	11	2		1
Immediate Action Procedures	9	6	7	2		
Body Positions	11	8	8	1	1	1
Sight Alignment & Sight Picture	14	9		2		
Trigger Control	13	9		3		
Grade how realistic the simulator was in the following areas:						
Zeroing:						
M16	12	8	11			1
M60	6	4				
M2	7	6	3			
Weapon Functioning:						
M16	12	8	2			1
M60	7	4		2		
M2	9	2	1			1
Tactical Scenario Simulation:						
M16	18	8				
M60	7	6				
M2	10	3				
Qualification:						
M16	12	6	1			7
M60	7	5	1			
M2	9	1	1	1		1
Rate the overall ability of the simulator to prepare you to conduct Live-Fire training:	12	10	3	1		
	YES	**NO**				
If you had access to a simulator would you use it to prepare for a live fire shoot?	24	2				
Do you feel the simulator gave you an edge when you conducted the same training Live-fire?	23	3				
Training ammunition is going to be cut. If we used 50% of our ammo $ to buy this type of simulator would the $ be well spent?	23	3				

Note (1): Class 12-93 had 26 machinegunnery students

Appendix B

School of Infantry (E)
Infantry Training Battalion Class 12-93
Basic Infantryman Simulation Questionnaire

	Outstanding	Good	Adequate	Needs Improvement	Poor	No Response
How well did the simulator meet the following areas?						
Safety	62	35	10	3	2	1
Loading	66	37	7	1	1	1
Unloading	59	41	8	2	1	2
Reloading	55	36	16	3	1	2
Rapid Reloading	54	38	12	5	1	3
Clearing	56	41	12	1	1	2
Immediate Action Procedures	67	31	9	0	2	4
Body Positions	50	34	21	5	2	1
Sight Alignment & Sight Picture	59	35	15	0	3	1
Trigger Control	59	31	18	2	2	1
Grade how realistic the simulator was in the following areas:						
Zeroing:						
M16	56	37	12	1	3	4
Weapon Functioning:						
M16	71	30	10	0	1	1
Tactical Scenario Simulation:						
M16	72	29	7	2	1	2
Qualification:						
M16	52	28	14	2	0	17
Rate the overall ability of the simulator to prepare you to conduct Live-Fire training:	48	44	15	5	0	1
	Yes	**No**	**No Response**			
If you had access to a simulator would you use it to prepare for a live fire shoot?:	107	5				
Do you feel the simulator gave you an edge when you conducted the same training Live-fire?:	104	7	2			
Training ammunition is going to be cut. If we used 50% of our ammo $ to buy this type of simulator would the $ be well spent?:	101	10	2			

Note(1): Class 12-93 had 113, Basic Infantry students

THE AN/SQQ-89 MAINTENANCE TRAINING EXERCISE: A LESSON IN THE TEAM APPROACH TO INTERACTIVE COURSEWARE DEVELOPMENT

Ken Fearn	James Gray	Senior Chief Sonar
Marconi Simulation	Marconi Simulation	Technician (Surface)
Dunfermline	San Diego	Danny Riley
Scotland	CA	USN

ABSTRACT

The paper provides an overview of the AN/SQQ-89 maintenance training program. It concentrates on the methodology used to provide one element of that program, the 'Maintenance Training Exercise'. The methodology relies heavily on the interaction between USN Subject Matters Experts (SME) and training managers and other specialists such as instructional designers, authoring system experts, graphics experts and software engineers. The paper describes how a USN SME and an Instructional Designer produced a prototype lesson. The architecture of the lesson is described as well as the tools which were used. The prototype lesson was validated in a classroom and the classroom and training manager feedback caused the prototype to be changed. The feedback is described and the changes it caused. The paper goes on to describe how the production process was automated to reduce the exercise preparation time from weeks to just days by a rule based approach. The paper concludes with a comparison of the effort to produce production lessons against the prototype lessons and a summary of the experience gained during the development.

ABOUT THE AUTHORS

Ken Fearn is a Consultant Engineer with Marconi Simulation, with 25 years experience in the development of software, mainly in real-time training simulators. Activities have included:- the Mandarin(TM) computer based training system, the design and development of an artillery sound ranging simulator and sonar simulators for both military and civil use, participation in the design and development of a parallel processor which was used to simulate an Advanced Gas Reactor in real time, and outside of the simulation sphere:- the design and development of an operating system and communications software for advanced banking terminals. He has published a number of technical papers, including three at previous I/ITSCs.

James Gray is a Senior Courseware Engineer with Marconi Simulation, with five years experience in the design and implementation of interactive courseware using the Mandarin system. James has been involved in several courseware projects encompassing a wide range of subject matter including Electronic Warfare and sonar. Other experience includes two years in training and authoring of Hypertext systems. James is currently based in San Diego working with the US Navy in a project to produce more than 800 hours of courseware for the AN/SQQ-89 sonar system.

Danny Riley is a Senior Chief Sonar Technician (Surface) with sixteen years active Naval service. In 1982 he earned the designation of Master Training Specialist. He is currently at FASWP assigned to the Curriculum and Instructional Standards Department as ICW curriculum editor. He is a graduate of numerous Navy technical and instructional systems development courses as well as the USAF interactive Videodisc Designers course. He has been a member of the AN/SQQ-89 maintenance ICW development group from its inception and is responsible for quality assurance and quality control of all lessons developed. He is intimately familiar with all aspects of the development process, from initial content outline and story boarding through authoring and on-line student presentation.

THE AN/SQQ-89 MAINTENANCE TRAINING EXERCISE: A LESSON IN THE TEAM APPROACH TO INTERACTIVE COURSEWARE DEVELOPMENT

Ken Fearn
Marconi Simulation
Dunfermline
Scotland

James Gray
Marconi Simulation
San Diego
CA

Senior Chief Sonar
Technician (Surface)
Danny Riley
USN

INTRODUCTION

Background

The AN/SQQ-89 Maintenance Training ICW (Interactive Courseware) project came about because of an increased demand for trained technicians. This was caused when the Navy installed its state of the art sonar suite on all the newer class ships and refitted on many other platforms. This put a high demand for student throughput at the school house.

Each class is approximately thirty four weeks long and has twelve training seats per class. Most of the students that attend the ITASS (Integrated Towed Array and Sonobuoy Sensors) course are junior personnel with limited shipboard experience. This course is designed to train that individual to perform intermediate level maintenance on a ship. This means a graduate of the course is able to identify and localize a fault to the board level. Student backlogs have started even running three shifts and multiple classes per shift. One reason for the long course length is the limited availability of technical training equipment time. Of the twelve students only three can be in the lab at one time. That means the instructor has to re-iterate simple procedures at least four times until each student can perform them. At the same time nine students are back in a class room in private study without an instructor. One way to get more people through the course is to shorten it. However, if you reduce the course length without maintaining the quality you end up with an unskilled technician lacking in confidence. ICW was seen as a way to reduce the course length and maintain the quality. Actual trainer time can be reduced by teaching 'knobology' and simple procedural tasks on ICW. That way, time spent in the lab on real equipment can be spent more profitably.

Team

The development team is a unique marriage of Navy, DOD civilian and contractor personnel brought together under the same roof to develop courseware for the Navy under the direct control and supervision of the Navy. The development team is managed by a DOD Instructional systems design management expert. It consists of Navy curriculum developers, subject matter experts and courseware authors as well as civilian instructional designers, courseware authors, graphic designers and software engineers.

To fully understand the uniqueness of this group it is worth looking at the more common courseware development process. Once a requirement for courseware is established, a contract is awarded to develop it. Typically the contract contains guidance on the content and structure and, in some cases, the development methodology. The contractor then goes back to the factory and develops plans and paper materials in support of the courseware, based on an interpretation of the requirement. These materials are then presented to the Navy for review. Comments are solicited and changes made, once again based on the contractors interpretation of the comments. This review cycle goes on until the acceptance of the plan and paper materials at which time on-line courseware development begins. Periodically during the on-line development phase, IPRs (In Process Reviews) are scheduled. An IPR is the first time the Navy really gets to see what it is buying. Again comments are made and the contractor goes away and make changes. This continues until the courseware is eventually accepted. And only then can the

process of validation begin. The Navy's involvement in the development is limited and really amounts to nothing more than providing the contractor with a list of things to fix.

By contrast, the Navy's role is central to this project. The team is divided into lesson development teams which consist of a team leader (Navy), instructional designer (contractor), SME (Navy) and a courseware author (either Navy or contractor). The members of a lesson development team work very closely on each step in the development process. The courseware author creates the treatment plan, storyboard and on-line lesson. This takes place under the watchful eye of the instructional designer who ensures the lesson is instructionally sound. In adition the SME verifies the technical content and provides Navy insight through shipboard experience. The team leader pulls it all together; scheduling reviews (external and internal) consolidating review comments, and ensuring Navy curriculum standards are met. This closely knit development team results in immediate feedback to the developer on what the lesson should contain. Technical errors are caught in the early stage of development. Money is not wasted on rework; it is done right the first time. Lesson development schedules do not stretch to eternity and review cycles take days not months. And most importantly, the end user, the school house instructor (who is currently teaching the course and will eventually use the courseware) gets to provide input, suggest changes and validate the lessons during development when it costs virtually nothing, instead of trying to change a finished product.

Lesson Types

There are three major types of lesson being produced using the Mandarin for Windows courseware authoring system.

ICW Tutorial

This strategy is used to present factual information of the sort the trainee needs to establish the foundation on which to build. This information could be the name of a switch, where it is located, what its function is in different positions and perhaps when to use it. The information is straightforward and never changes. This type of information is traditionally taught as chalk and talk with the trainee looking at either a line drawing view graph or a picture in the technical manual. First, all the switch names would be covered, then all the different positions followed by the function and so on. The instruction was fragmented at best. All the pertinent information was covered, but not at the same time. When the students went back to study their notes they would be flipping pages back and forth trying to put it all together. In our ICW tutorials we have consolidated information. For example, all the prerequisite information the student needs to operate the switch is presented at the same time as the student is looking at the actual switch. The information is presented in audio, backed up with text and graphics. The student controls the pace, takes notes and reviews as necessary. In short, the tutorial lessons are used to present knowledge the trainee will need later to perform some task.

ICW Integrated

The integrated lesson is used to present knowledge and develop a skill at the same time. It differs from the tutorial in that the knowledge portion is now procedure specific. The lesson itself centers around performance of a procedure. To enable the students to properly perform that procedure, the why, what, when, where and how is covered, and then followed by "now do it". This is done for each step until the student has completed the procedure. This type of lesson did not exist previously in the course because it requires one on one instruction. It did however exist in the fleet in the form of 'on the job' training. The new 'strikers' (rookies) were paired up with experienced technicians until they knew what they were doing. It was very effective; the new technician learned exactly how to do something and it proved more cost effective because less things broke during adjustment. Development of the integrated lesson enabled us to go 'one on one' with the student, teach the knowledge and associated skill simultaneously while providing guidance based on the students interactions. The more knowledge the student exhibits, the more the lesson moves towards performance.

ICW Exercise

The exercise lessons allow the student to use the knowledge gained in tutorials and practice the skills acquired in integrated lessons. We really have two different forms of exercise lessons. The first is based on step by step procedures for adjustment, alignment, equipment tests and so forth. In this type of procedure each step is completed sequentially with an expected outcome. The second type of exercise is based on fault diagrams and is used to troubleshoot equipment casualties or malfunctions. In this case the trainee configures the equipment in a known state, makes observations and then decides what to do next based on those observations. All exercises start with an exercise briefing. The student is told what is wrong and what needs to be done. From that point on the only instructions or prompts the student receives are in the form of feedback when a mistake is made. The difference between the two types of exercises becomes evident when you examine where, when and the type of corrective feedback the student receives. For instance, in the step by step procedure you always know what the next step is and exactly what you have to do in the current step to move on. So it is very simple and the developer writes feedback specific to a single step and only that step. In this case, when the student makes an error it can be immediately detected, but in the exercises where the student is required to make decisions based on multiple observations it becomes more difficult. First the student must make various switch settings to configure the system correctly, then make the required observations to make a decision. Only when the student makes his decision can an error be detected. This latter type is the most difficult to develop because there are so many different possibilities that writing feedback for any one may give away some other aspect of the problem. Since most of the trouble shooting procedures in the course involve fault logic diagrams or fault trees, development of this type of exercise became a prime concern.

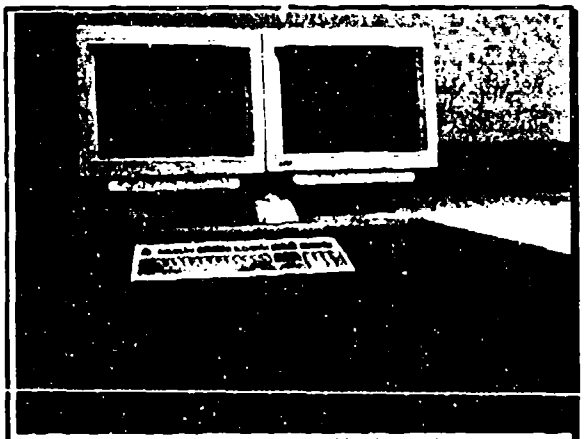


Figure 1 Student Workstation

THE DIAGNOSTIC FAULT TREE

The maintenance philosophy for the majority of sonar maintenance courses has evolved to fault diagrams. The complexity of the systems and data transfer speeds resulted in 'black box' trouble shooting. That is to say the technician no longer chases electrons from pin to pin to locate a problem. Now the technician follows set procedures to ensure the system is configured properly, then makes initial observations as a basis to answer YES/NO questions. The answers determine what path the technician will follow in the fault tree. This process of making settings, observing the outcome, answering a question, continues until the faulty component is located. This makes understanding how to read and interpret the fault tree crucial to becoming a good technician. Following a fault tree block to block in itself is not a difficult task. What complicates the task are the associated notes, references to other fault trees and procedures, where to get the information or make an observation, how and where to make equipment settings. If the student is familiar with the equipment and can follow a fault tree he can locate the problem. That is why teaching the process of trouble shooting rather than a particular fault tree leads to a better technician.

The remainder of this paper addresses our technical solution to teaching maintenance using fault trees.

THE PROTOTYPE

The objective of the exercise lesson is to present a troubleshooting diagnostic procedure which incorporates the fault tree from the sonar technical manuals. This allows the student to practice troubleshooting a sonar fault and also links the process to the technical documentation. In order to carry out this task the student has to be given a view of the current step of the fault tree and access to all sonar equipment onboard the ship. The exercise has to be technically correct, easy to use and also encourage the student.To complete any one section of the fault tree the student may be required to gather information from several different pieces of sonar equipment. These are scattered throughout the sonar spaces on the ship, so consequently the student has to have the ability to move freely from one room to another observing data and settings.

The process involves the student setting the equipment into various states by entering values into data entry fields. He then observes readings and compares them with expected values. As this task is performed the lesson automatically notes the student's verifications or settings, and marks them off if they were appropriate to the current fault tree step.

The student can attempt to proceed to the next fault tree step at any time. If he has gathered all the correct data then he is given the 'correct' feedback and allowed to progress to the next step. However if the student has not verified all the necessary data then he is given one of three negative feedbacks dependent on the number of attempts. These take the general format of :

(1) 'No, you have made a mistake, try again.'

(2) 'No that's wrong, here's a hint.'

(3) 'No this is what you should do. Now do it.'

The student is awarded points for each completed block of the fault tree, with more

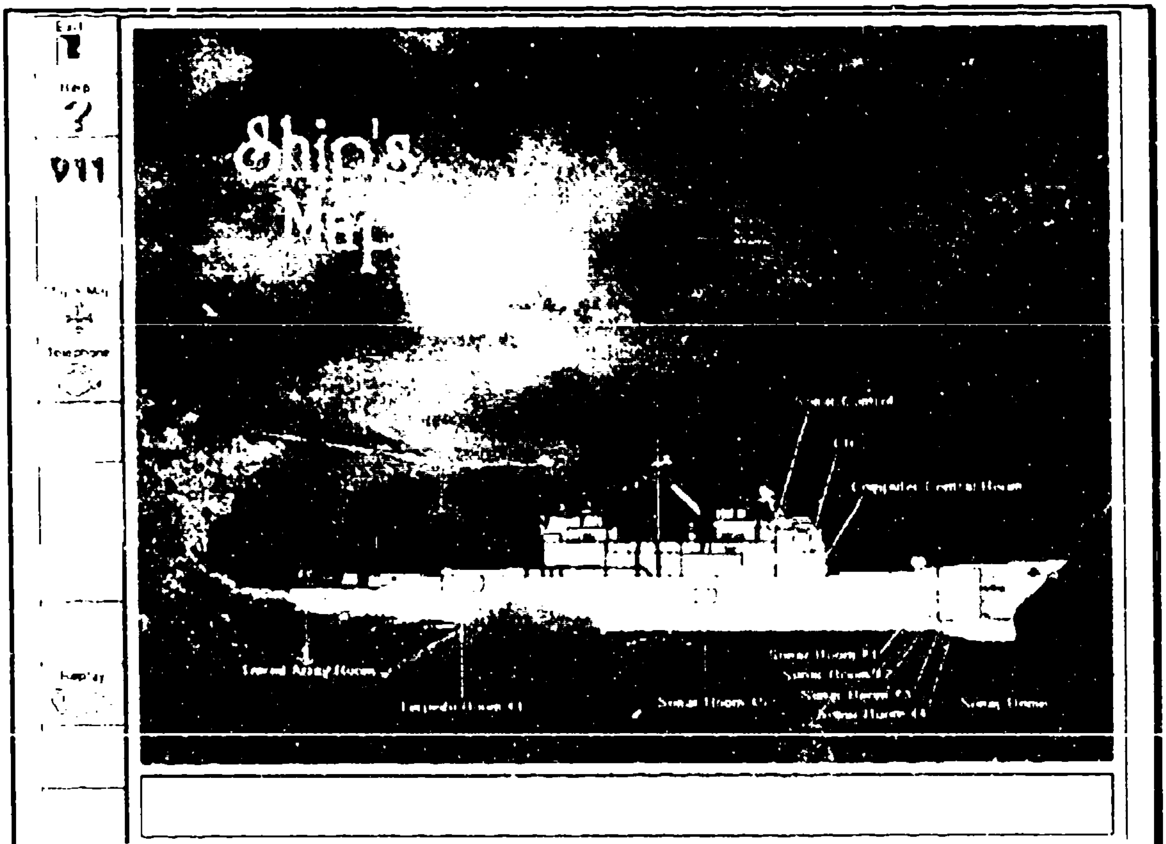


Figure 2 Screen One

points for fewer attempts. The student is also timed throughout the exercise, scoring more points for a speedy solution.

Student Operation

The lesson is designed to be delivered on a dual high resolution screen PC system (see figure 1), with the student's main input through a trackball. The following outlines the various display areas.

Screen One

Screen one (see figure 2) displays a control bar (at the left of the screen) which is always available to the student, and a status bar which gives feedback of the exact location of the student and a main display area which shows either a map of the ship or the piece of equipment the student is currently interacting with.

The control bar displays a series of icons always available for use at any point in the exercise. Figure 3 shows two of particular interest.

Selecting the Ship's Map icon displays a map of the sonar spaces that the student technician can travel to. The movement between spaces is depicted by an animation to remind the student that valuable time is being taken.

Selecting the Telephone icon displays a menu of compartments the technician can call and a choice of questions he can ask. This facility avoids unnecessary travel and is more in line with real operation onboard ship. Traveling to another sonar space is valid but eats up valuable time and can reduce the student's score.

Screen Two

Screen two (see figure 4) displays the current block of the fault tree the student is attempting to complete, the student logbook (where verifications and settings are logged) and the main display area shows the current

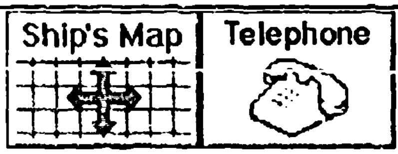


Figure 3 Control Bar Icons

sonar space the student has entered. There are three main display areas on the second screen.

Fault Tree

This displays the current section of the fault tree for the student and takes the form of either.

(1) A list of data the student must 'verify' or 'Set' on various sonar equipment and an 'OK' button to select after completion of a step.

OR (2) A question, and a list of data to 'verify' or 'Set' along with 'YES' and 'NO' buttons to answer the question.

Current Sonar Space

The second screen always displays a graphic of the current sonar space the student entered. Each cabinet of equipment displayed within this space is selectable.

When a cabinet is selected it appears on the first screen. However, the sonar space display remains on the second screen for the student to select another cabinet.

Student Logbook

The student navigates around the ship attempting to verify information to complete the current fault tree step. As the student verifies or alters a current reading the data is automatically recorded to the student logbook. This gave the student a record of data recorded and can be used later in the exercise.

The student can record any data he wishes in the logbook, however only verifications appropriate to the current fault tree step enable the student to proceed to the next step of the fault tree.

To verify a piece of data the student clicks on a control or indicator and then the name of the data field and its current status is entered into the student logbook (as well as being checked that it is appropriate for the current fault tree step).

As the student progresses through the exercise his logbook has a record of information previously verified. These 'notes' can be re-used by the student to answer future fault tree steps.

PROTOTYPE VALIDATION

The development strategy and process utilized tc produce the prototype lessons had a comprehensive review cycle built in. At each stage of development the end users were consulted to ensure what went into a lesson was accurate and really needed to be there. Involving the instructors as subject matter experts was seen as one way to reduce costs and development time and provide them with an avenue to influence course content. They were already familiar with the curriculum; they dealt with the target audience on a day to day basis, and more importantly, they knew what worked and what did not when it came to getting the point across. With this constant involvement, technical errors in the prototype were limited and the user interface evolved to its current state. The prototype lessons were presented to groups of ITASS students in various stages of training as well as operator and maintenance instructors. Several interesting things came to light. The instructors had more difficulty performing the troubleshooting exercises than did the students. The instructors wanted short cuts incorporated that the system and curriculum standards would not allow. The students on the other hand followed the procedures in the technical manual and consistently scored much better.

The prototype allowed the student to proceed down incorrect paths of the fault tree before negative feedback was given. For example, for a particular fault tree the student may be required to verify six separate items of data, then select 'OK' to proceed. The prototype allowed him to proceed even if less than six items were verified. The negative feedback was not given until the student made a decision on this new step at which time the student was returned to the step where the original error was made.

This strategy sounded fine in the planning stages but, when put in front of the student, the reality lead to frustration and lack of motivation. The exercise was changed to give

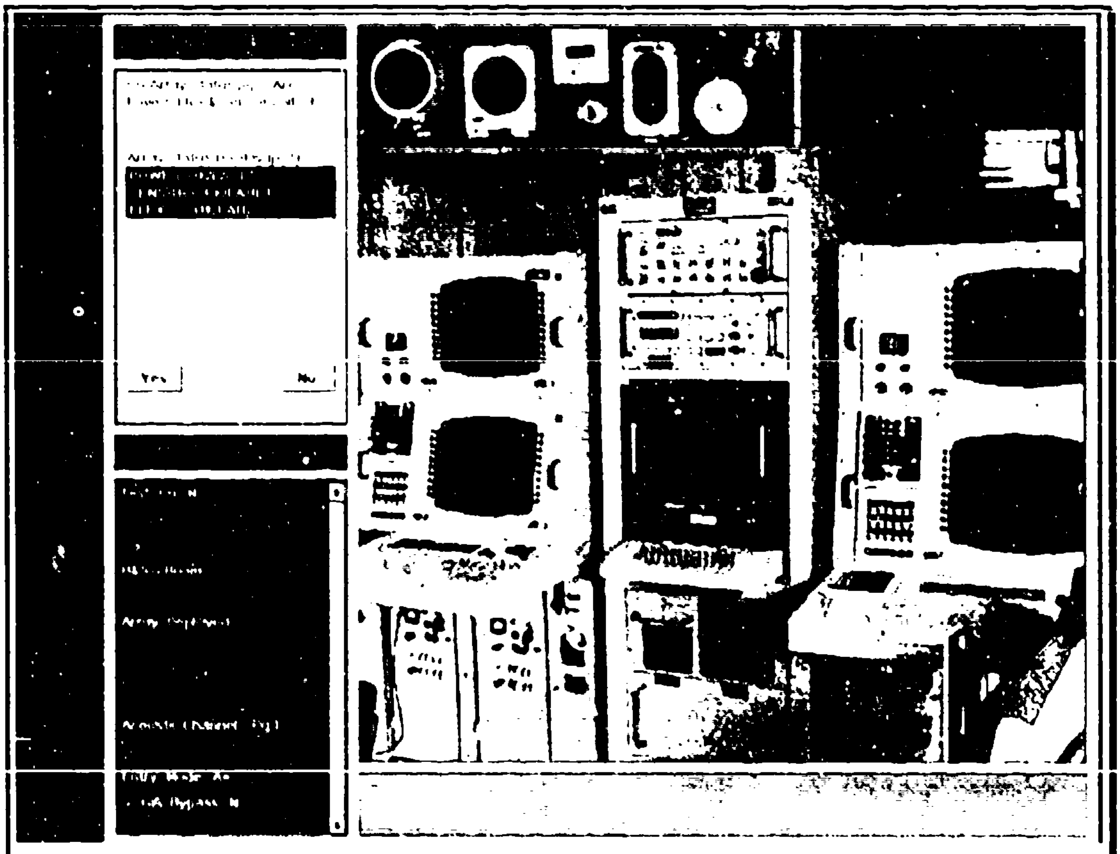


Figure 4 Screen Two

feedback as soon as the student made a decision at any fault tree step. This change of strategy was an important factor in the design of the production lessons, dramatically reducing their complexity for an increase in training efficiency.

THE PRODUCTION MODEL

Having established the training value of the prototype, a closer look at a production version was needed. In the prototype system, developing a new fault finding exercise was expected to take over four weeks. In order to achieve the productivity targets this would have to come down to less than a week. The entire prototype model was written in the Mandarin courseware ...uction language, IVL, which had proved ...n efficient prototyping tool. Each part ...tctype model was examined:-

...cography Model

This is generic code for most ship types and so the prototype could be used directly in the production model. The code can be used for most ship types because of the use of Mandarin's pickboxes. These are invisible areas placed over the pictures. A touch or mouse click in these areas returns the name of the area. So a change from one ship type to another actually only involves a change of pictures, provided the picture name and its pickbox names remain constant.

Equipment Model

We are only dealing with the AN/SQQ-89 sonar system, and throughout the exercise the student is interacting with various components of this system. The authoring language was used to develop modules which emulated the individual pieces of equipment that made up the overall system. These modules were developed for the prototype

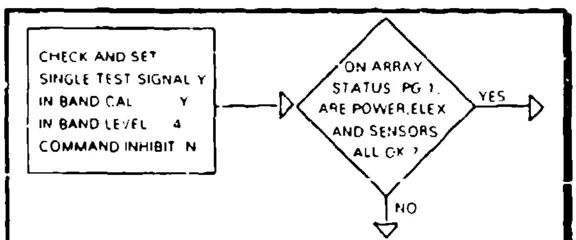


Figure 5 Fault Tree Example

and are now directly used in the production system.

Fault Finding Exercise

This is where the production savings were expected to be made. The prototype codes the exercise as one entity. Separation of a fault finding exercise into a fault tree and the fault data would clearly achieve some benefits by allowing the fault trees to be re-used over many exercises with different fault data.

Fault Tree Interpreter

Perhaps the most interesting change was in the handling of the fault trees. These are no longer programmed into the lesson but are declared as separate data files. The AN/SQQ-89 fault trees consist of two types of diagnostic step (see figure 5). A 'procedure' step (rectangle) contains a list of actions to be performed (typically equipment settings) and has a single exit. A 'decision' step (diamond) contains a list of questions (typically comparing of data with expected results) and a 'yes' and a 'no' exit.

The fault tree is declarative and is independent of the actual fault data. In the prototype, only the route through the fault tree for the current exercise was needed. For the production system, the entire fault tree is declared but without a path through. So the fault tree declaration reduces to a static coding of the fault tree flow chart, encoding the contents of each step and declaring the inter-connections, but without relating it to a particular fault condition. Figure 6 shows an example of the declaration of a single step.

The fault tree interpreter manifests itself to the student as a window displaying the state of the current step of the diagnostic procedure. Each diagnostic step has a title which is displayed to the student, plus the list of actions, which can be optionally shown to the student. As the student performs the required actions they can be highlighted in the window. If the student attempts to leave the step without having collected the necessary data, feedbacks are displayed to him. These are graduated so that more information is supplied each time an error is repeated, starting from a general warning and progressing to a precise diagnosis of the

student's problems.

A student communicates with the fault tree interpreter in two ways. The items in each step can be highlighted by simply observing the equipment or referencing notes in the on-line logbook. For example, if the fault tree says 'Check Power' the student only has to use the geography model to move to a picture showing the necessary control and then click on that control. Each control has a unique name encoded on an invisible pickbox superimposed on it. This name is also encoded in the fault tree declaration, so correlation between controls and fault tree steps is automatic without any explicit courseware action. Alternatively the student can gather the state of various controls into his on-line logbook and uses his notes to check off items in the fault tree step.

When a student judges a step to be complete he clicks on the 'OK' for a procedural step or on the 'Yes' or 'No' for a decision step.

Exercise Data

The original fault data was compiled into the lesson which meant that each exercise was a different piece of lesson code. For the production system, the fault data was defined in a separate file which was read by a lesson initialization module. So that the fault data definition file and the compiled lesson could address data by name a new Mandarin library was created which permits data to be directly addressed by name. This has the added benefit of permitting runtime debugging of

```
STEP 7-94-2a1, On Array Status pg 1,
Are Power, Elex & Sensors all OK ?;
Instruction Array Status Display (pg1)
CHECK power,     POWER     OK/FAULT
CHECK sensors,  SENSORS OK/FAULT
CHECK elex,          ELEX
OK/FAULT
FAULT 0,10,,posfb_.wav,
FAULT 1,8, roy1_ok.bmp,negfb1_.wav,
FAULT 2,6, roy3_lok.bmp,0500002_.wav,
FAULT 3,4, roy5_mad.bmp,0500003_.wav,
NO GOTO, 7-98-1b2
YES GOTO, 7-98-1b3
ENDSTEP
```

Figure 6 Fault Tree Declaration

the exercise using the same names.

The exercise data also contains a declaration of the route through the fault tree for this particular exercise. This simply lists the steps which will be visited and the correct exits from each step.

The changes to data handling meant that some minor changes to the equipment model were necessary to access the new exercise data rather than the old compiled-in data.

Fault Tree Exercise

All the fault tree exercises use precisely the same lesson code plus two additional text files to define the exercise data and the fault tree.

Performance

The production system has a slightly neater student interface for the fault tree window, and response times are slightly better. Management is clearly a lot simpler because of the need for only a single piece of lesson code. However the real benefits are in productivity gains. A new fault tree can be encoded in less than a week and re-used in many exercises with different fault conditions. A new exercise can be built around a fault tree in under a day. This represents at least a four fold productivity increase over the original prototype.

A further minor improvement was possible by creating WORD templates and macros so that the declaration files can be created a little more simply with less possibility of syntax errors.

An interesting side effect of the Fault Tree Interpreter is that is has immediate application to simple procedural training without the complex branching inherent in a fault tree exercise. It is now used in all ICW exercises carrying its production benefits into more areas than originally expected.

PRODUCTION VALIDATION

The production lessons have gone through the same extensive review process as the prototype lessons. Incorporating the comments, and improvements suggested by the instructors during review of the prototype lessons has resulted in fewer comments

during production review. Student frustration was reduced by implementing the feedback immediately upon the student's exit from a particular fault tree block, rather than the prototype technique of providing feedback when the student completed the next block. The user interface was improved: a single tiered control strip, which is now used throughout the course in all lessons, puts the controls at the student's finger tips and encourages their use. Comments from student and instructors that reviewed the original prototypes, as well as those who only saw the production lessons, have been enthusiastic. Students and instructors alike are eagerly awaiting the next batch of lessons.

CONCLUSION

There are a number of conclusions to be drawn from this experience.

Teamwork

The unique structure of the development team, and location of the development site, has had a direct influence on the success of this project. Bringing together a mixed discipline team of subject matter experts, instructional designers, graphics artists, software engineers and instructors from different backgrounds, different employers (and speaking different brands of English) under the direct supervision and management of the customer, in this case the Navy, has created a unique ICW production environment. It is difficult to see how any one of the groups represented on this project (DOD, Navy, Marconi, and other contractors) could have achieved as much as this highly motivated and well-integrated team achieved working together. Locating the development team at the school house enabled the Navy to provide constant and concise input on a day to day basis to the development team on lesson structure and content through all stages of development. This influence on what went into the product during the early stages resulted in fewer changes during the later stages when they are very costly. Having uniformed Navy performing the functions of quality assurance, quality control and enforcing instructional standards from within the team and also having a DOD Supervisory Instructional Systems Specialist manage the team has proven that communication and teamwork are essential to success.

Tools

In this instance the choice of tool allowed a growth from a concept-proving prototype to a highly cost-effective production system. In this instance, Mandarin for Windows provided a tool which provided the advantages of a courseware development environment along with the freedom to connect to a more general purpose software development environment. On the one hand Mandarin was designed to provide lesson structures and the associated student navigation within a ready built framework for a minimum of effort. On the other hand, it was designed to be extensible through its IVL programming language and through Dynamic Link Libraries. The flexibility inherent in Mandarin was critical, permitting the development of the very efficient specialized fault tree interpreter functions for this particular task and permitting them to be easily connected into the training framework.

Testing

Proving the lesson principles with the prototype allowed problems to be sorted at an early stage. If this had not been done until many lessons had been produced, the rework costs could have been large. As it was, rework was limited to the prototype, and the lessons learned allowed major savings on the production of the following lessons through simplification of the requirement and the more automated approach.

Summary

The keys to the production of successful courseware would appear to be Teamwork, Tools and Testing. All of these factors have to be brought together to produce the quality training materials that are required to meet the Navy's demand for trained sonar technicians.

PART-TASK COMPUTER BASED TRAINING FOR OPERATORS USING A TOOLBOX APPROACH

Commander M G Pengelly, Royal Navy
Lieutenant Commander D W Dowsett, Royal Navy
The Royal Naval School of Educational and Training Technology
HMS NELSON
Portsmouth, England

ABSTRACT

The focus of this paper is the development of a "toolbox" computer based training system, which is used to provide Operations Branch Ratings and Seamen Officers with part-task training in the use of the modified Action Data Automated Weapons System (ADAWS) coming into service in Royal Navy ships. This CBT is a networked, multimedia system which provides a limited simulation of the actual system within an interactive training lesson, which itself is based upon a scenario developed by the instructor.

The paper discusses the development of this system in the wider context of CBT development and examines the significant milestones in the development of the Royal Navy's toolbox strategy. It traces the analysis of training need and how technology can be applied to meet established objectives whilst offering a degree of control to the instructor. The paper concludes by reviewing how lessons learnt are being incorporated into a future CBT procurement for the Surface Ship Command System (SSCS).

ABOUT THE AUTHORS

Michael G PENGELLY is the Commanding Officer of the Royal Naval School of Educational and Training Technology (RNSETT). His primary task is to train Naval Officer and Ratings in the Systems Approach to Training and Instructional Technique. He also has responsibility for Research and Development in the areas of Technology Based Training and Multimedia Applications. Prior to taking command of the RNSETT in 1991 he was a member of the Admiralty Interview Board responsible for the selection of young Officers for entry into the Royal Navy. Commander PENGELLY has many years experience in Naval training and has served in the Ministry of Defence (London) and the Command and Staff College, Nigeria.

Dennis DOWSETT is the Training Media Development Officer in the RNSETT. He has wide experience of managing training execution in naval training establishments. He joined the RNSETT in 1990 having previously led Training Design teams at the Nuclear Training Group HMS SULTAN, and the Royal Naval Submarine School HMS DOLPHIN. He graduated from London University in Physics and completed a MSc in Material Science. He is currently studying Computer Science with the Open University.

PART-TASK COMPUTER BASED TRAINING FOR OPERATORS USING A TOOLBOX APPROACH

Commander M G Pengelly, Royal Navy
Lieutenant Commander D W Dowsett, Royal Navy
The Royal Naval School of Educational and Training Technology
HMS NELSON
Portsmouth, England

INTRODUCTION

There is an essential paradox confronting the Royal Navy (RN) in its policy for Computer Based Training (CBT) materials. For reasons previously discussed[1], the production of bespoke solutions to identified training needs is the norm for this type of training material. Nonetheless, it is an inescapable fact that many of the most successful and enduring CBT materials in the RN have originated from ideas developed locally by instructors. The reasons are not difficult to establish. Delivery of basic, first level training for which CBT is commonly employed requires a structured development with regular feedback in order to build trainee confidence. Although second nature to the instructor, this is not necessarily the case for hardware and software engineers whose priorities are likely to be very different. The "Joint Services Guide to Computers in Training" [2], identifies the following skills as vital to the production of successful courseware:

- Project Management skills
- Subject Matter expertise
- Course Design skills
- Graphic Design skills
- Programming or Authoring skills

It is inconceivable that the RN could justify deploying its own servicemen to meet theso varied requirements for any sustained period. With the inevitable reductions in manpower that are now taking place, this situation will not change. The Navy does however, recognise the problems that this creates: difficulties in specifying precisely its intended requirements, the "remoteness" of the instructor to the training materials and the lag times in responding to changes in operational software or procedures. The solution that the Navy began to explore in the mid eighties was the use of software toolboxes designed to meet specific training needs. This paper traces these developments and describes in detail the system devised to provide basic training to Seaman Radar picture compilers for the Action Data Automated Weapons System (ADAWS).

ORIGINS OF INSTRUCTOR TOOLBOXES

The development of CBT materials within RN establishments goes back to the late seventies and introduction of the first serious and widely used, disk operating home computer in the United Kingdom, the BBC B. This computer used a version of BASIC as a programming language and was extremely versatile. its graphics capability was surprisingly powerful considering its limited memory. Promising local developments were supported where appropriate and a Morse code trainer and a sonar aural analysis system (multimedia way ahead of its time !) still survive to this day in various guises.

The advent of the PC and the introduction of authoring systems such as TenCORE, Mandarin and Mentor sustained these local developments for some time. However a

CBT industry was by now growing up in the UK and when the Ministry of Defence (MOD) commissioned a study into CBT in 1985[3] it was apparent that CBT must be placed on a more professional foundation if it was to develop further. The Navy established a Lead School for CBT, provided training and developed a procurement strategy for its solutions. [4] Recognising that successful implementation needed the involvement and commitment of its training staff, the Navy began the development of two major toolbox initiatives: Sonar Analysis Continuation Training and a Bridge Watch Keeping package "Rule of the Road".

SONAR SKILLS TRAINING

The Royal Navy's submarine flotilla issued a software toolbox requirement to build test Lofargram simulations for its sonar operators to analyse. This was to form a key part of its Continuation Training programme, designed to preserve perishable skills amongst its sonar operators[5].

In essence the system allows instructors to create and deliver to students on a PC system a Lofargram simulation of almost any sonar effect. Being largely equipment independent, it enables this training to take place at relatively low cost, at widely distributed training sites and even at sea. Hitherto this type of training had been conducted using materials gathered from operational patrols. Parameters are passed by the instructor to an underlying mathematical model in order to create a scenario of the desired complexity. It is perhaps more accurately termed "computer based testing", as the feedback provided by the system is intrinsic. It does however, demonstrate how the facilities available to an instructor can be significantly enhanced, whilst dramatically reducing costs using a computer based approach.

RULE OF THE ROAD

The Rule of the Road courseware was developed for the Royal Navy by Hughes Rediffusion Simulation to train Bridge Watch keepers in the correct procedures for handling potentially dangerous situations at sea. Although reference materials that include lights, sound signals etc. are available to students, the core of the courseware is the scenarios, developed by the instructors.These scenarios enable students to run what appear to be real time simulations which, on the student monitor, can be observed either as bridge views or radar plots Figs 1 & 2. In creating scenarios instructors have control over visibility, vessel type, number, sea environment, track and speed Figs 3 & 4. Scenarios can be carried out as "hands off" Demonstration, Practice scenarios providing help, guide and freeze facilities, or Assessment scenarios where student actions can be downloaded to disk for instructor evaluation.

Rule of the Road is not a real time continuous simulation in the sense of the previously described sonar analysis software, but a discrete simulation that contains extrinsic feedback to students relevant to the actions they have taken. The apparent real-time sequences put pressure on the student to make decisions within situations created from the toolbox.

THE FINDINGS

This type of solution struck a chord with trainers and several more such projects have developed from drawing board concepts, to fully fledged projects, of which ADAWS is just one. The benefits that this approach has provided are seen as three-fold. Firstly, it promotes "ownership" amongst the trainers. If something is not quite right within a scenario it is within their power to make changes. This means that they can respond to trainee needs on the basis of experience rather than being forced to follow design work carried out long before they moved into post. Secondly, it overcomes the technical inhibitions of the staff, particularly the older ones. Thirdly, it avoids the bureaucratic problems of contractual updates that frequently leave training materials lagging behind the operational software.

Set against these benefits it does impose

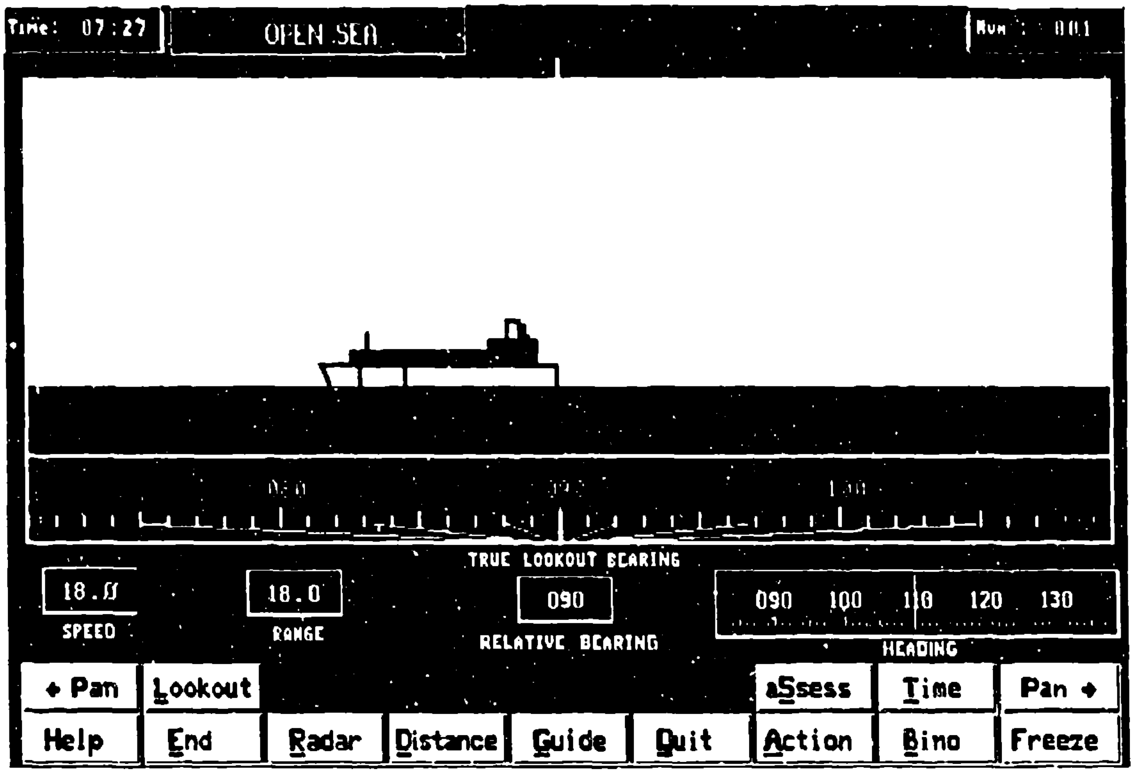


Figure 1 Bridge View

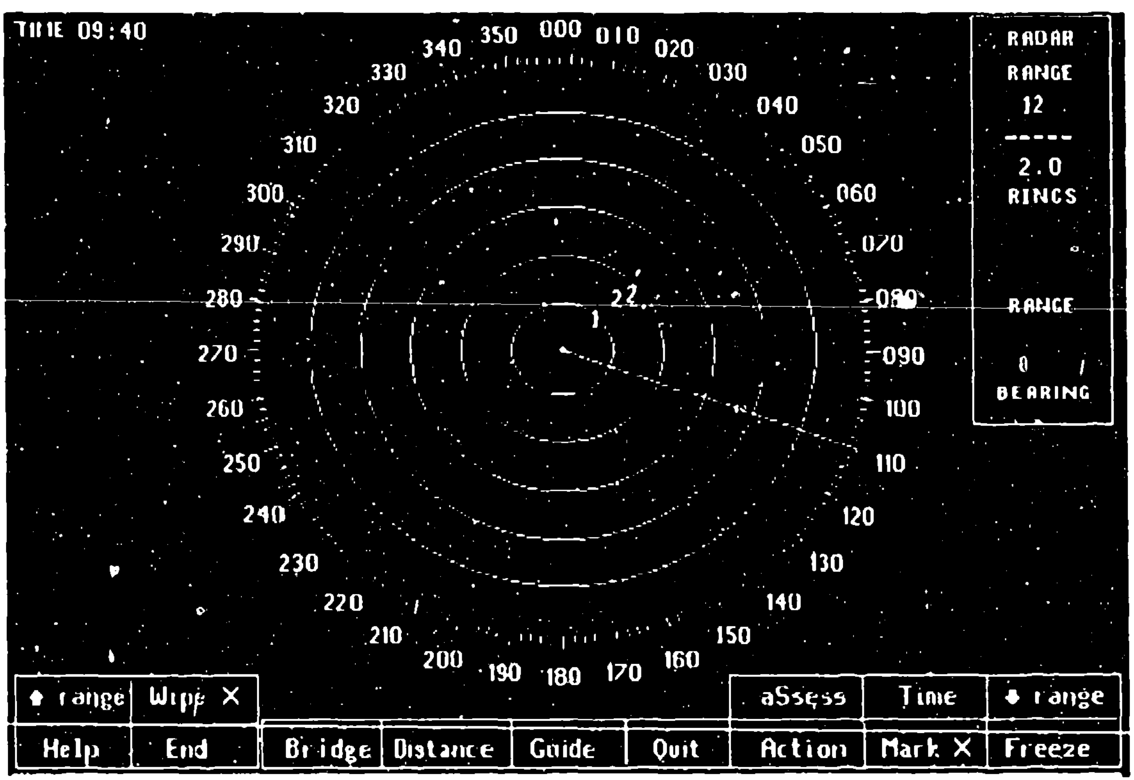


Figure 2 Radar View

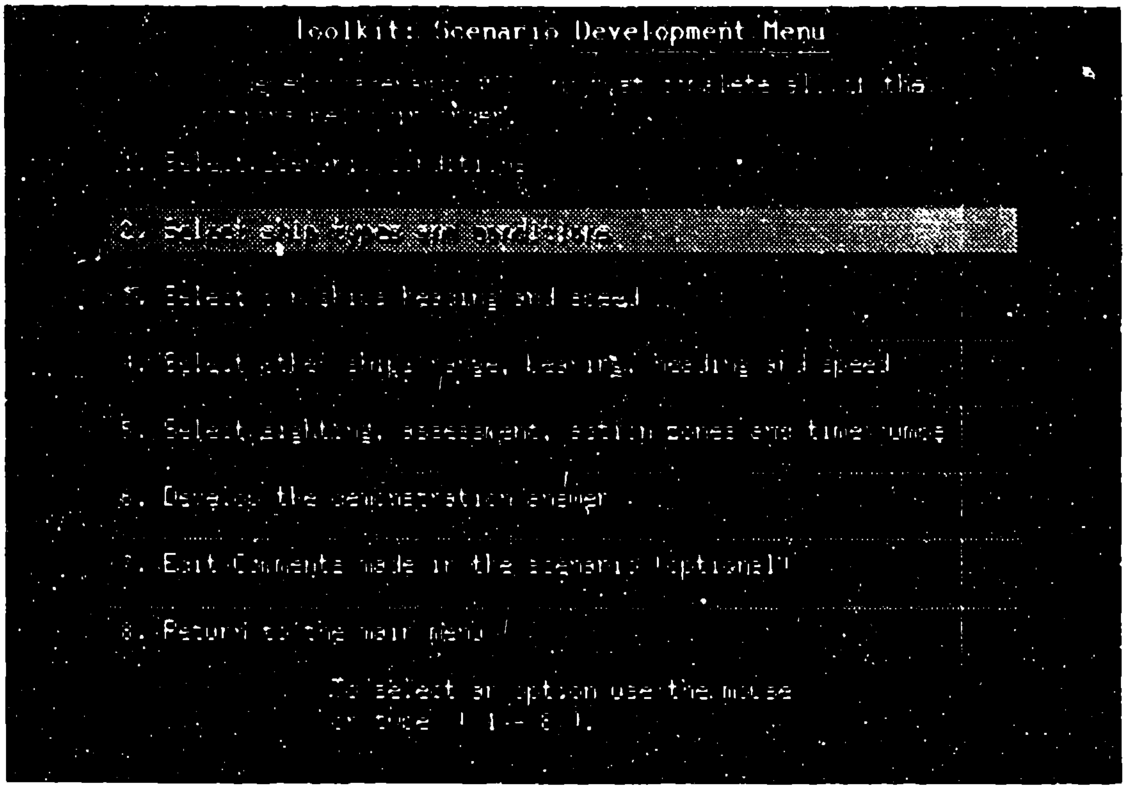


Figure 3 Instructor's Toolkit Menu

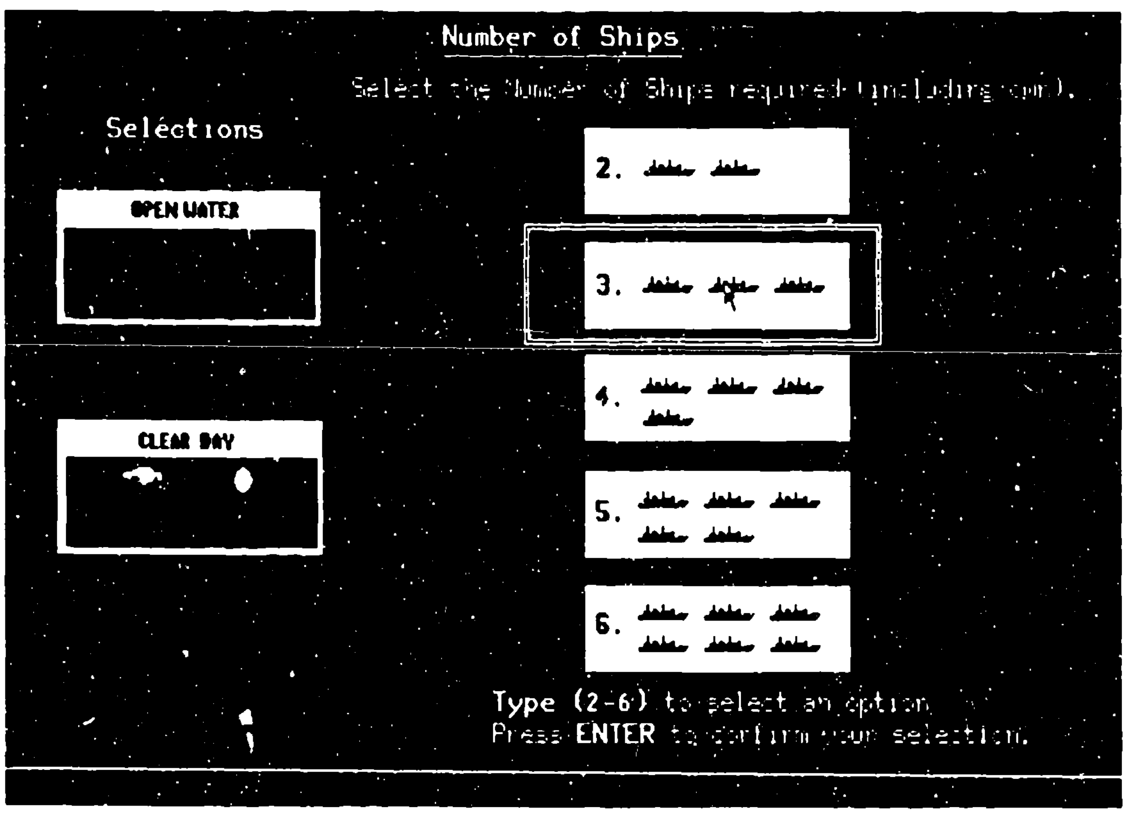


Figure 4 Number of Ships Selection Screen

an additional burden on the establishment's Training Quality Organisation as they are responsible for the standard of material presented to the student. There is also the danger that instructors lose sight of, or are unaware of, the original objectives addressed by these solutions. They endeavour to employ the tools for higher level objectives and become frustrated or disparaging when they are unsuccessful.

The design effort required to specify these tools is also considerably greater than for more standard CBT. The software design decisions necessary for this type of courseware mean that the costs of subsequent modification can be extremely high. Not withstanding the concept has proved very sound.

MEDIA SELECTION

The success of the systems described should not really have come as any surprise. The RN operates a systems approach to training that calls for job and task analysis and a design process which relates these analysies to its training and enabling objectives. Both of the requirements described followed this philosophy and the media selection processes were cognisant of the learning outcomes required. These concepts are applied mindful of the need to "analyse both the reality and the objectives"[6]. The identification of the simulation fidelity level resulted from:

a. Analysis of the real skills, phenomena and situations that have to be learnt by the trainee; and

b. Breaking down the learning tasks and assessing the degree of difficulty in order to describe what degree of reality abstraction is acceptable.

It was recognised that discrete and continuous simulations such as those so far described could significantly extend cognitive learning to exercise logical thinking and introduce elements of reactive learning. It was also very apparent that the rapid increase in computing capability, the availability of a wide variety of user interface and the emerging multimedia technologies could considerably enhance the instructors ability to address both the psychomotor and interactive domains. [7]

ADAWS MOD 1 CBT REQUIREMENT

ADAWS is fitted to the Royal Navy's Type 42 Destroyers and CVSs. It is the vehicle by which the ship interprets the responses of its sensors and controls its weapons. It is therefore essential that every member of the team is trained to a high degree of proficiency in this system as the security and effectiveness of the ship depend upon it. This system, which has been successfully operated for some years, has undergone an extensive upgrade, referred to as ADAWS Mod1. This will be phased into service gradually. In 1989 a methods and media analysis was conducted for the training needs associated with its introduction. There were several issues relevant to the final media decision.

a. The ADAWS system was well established and although Mod 1 significantly changed the way the system was operated the underlying philosophy remained the same.

b. Realistic team training was already being carried out in a fully simulated operations room.

c. The Mod 1 system had not been accepted into service at the time of the analysis and although changes at that point were likely to be minor, if the training was to be in place on time a decision could not be delayed.

d. The training organisation involved had used computer based training for many years and there was unlikely to be resistance of a cultural nature.

The computer based training materials already being used were very much first generation and largely tutorial in nature. The level of training outcome identified by the analysis indicated that the competencies required on completion of training demanded more from the media. Knight and Morgan [8]

proposed a workstation based Integrated Training System (ITS) which they proposed could replace traditional training patterns Fig 5.

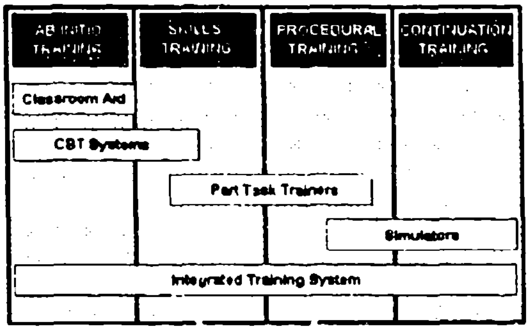


Figure 5

The ADAWS team were sceptical that the fidelity required for the later stages of training was yet achievable in this fashion and the existence of a high fidelity system in this case made such an approach unnecessary. The team did believe that at low risk and moderate cost it would be possible to extend CBT solutions to meet high fidelity simulation in the training spectrum Fig 6.

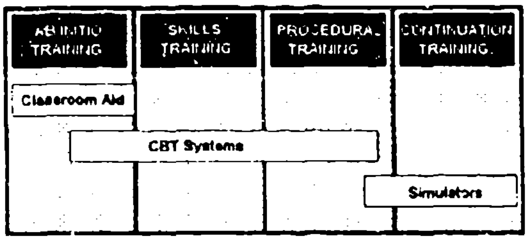


Figure 6

The design goals that the team strove for in producing their specification were:

a. It would be capable of meeting the training objectives for the initial phase of training.

b. The system would behave as an "electronic chalkboard" which enabled instructors to use their skill and experience to develop scenarios which would unfold in front of their students. The representation of the scenarios would be as realistic as possible and demand actions or decisions from the students, either intrinsically or extrinsically.

c. The scenarios would not be complete simulations but would allow only limited deviation from the solution originally envisaged by the instructor.

d. The instructor would be able to provide relevant feedback through the software.

e. The system would be easy to use and not require specialist computing skills.

f. The delivery system would be PC based.

g. It would be capable of modification by the establishment staff should changes occur to the operational software or the operating procedures.

h. The time for a typical scenario build should not exceed two man days.

i. The system would use a representation of the actual ADAWS keyboard as its primary input device. The training would be centred around its use and the scenarios unfolding on the monitors.

j. The instructors ability to generate tutorial materials would be limited.

ADAWS MOD 1 - CBT SYSTEM

The contract for the ADAWS CBT System was awarded to MSC Vosper Thorneycroft (UK) Ltd in the Spring of 91 and delivery was competed in Feb 92.

The system is a Novell Network of 386 PCs: 12 student stations and 1 instructor author station. The PCs have twin 14" VGA display screens and realistic ADAWS facsimile keyboards together with 3 button tracker-ball devices for their inputs. They also include digitised sound delivery boards which provide audio prompts or commands over headphones. The instructor station also has access to a laser printer for the downloading of screen display hardcopy and a digitising camera to scan in images where

necessary. One screen represents the surface radar plot whilst the other represents the tote pages of information.

The key software deliverable was the instructor's scenario generator which enables the instructor to compose scenarios of varying complexity according to the needs of the trainee. The training materials are built up using menus, icons and simple text inputs. To assist the instructor the system contains libraries of data for his use. These include:

a. Injections - these are made up of letters, numbers, spaces, hyphens, plus, query, inject and erase. In the actual system there are over 800 injections.

b. Symbology Track Data - these will be made up of combinations of letters and numbers.

c. Voice - Digitised voice reports or commands.

d. Tote pages - these tote pages contain the information to which the student must respond. The data on the tote pages must correlate with what is seen on the screen.

e. Marker and Tactical Areas - these include conflict and tactical areas, weapon safety, detection and tracking windows. The markers move with tracks.

The tracks are created by the instructor completing a matrix similar to that shown in Fig 7. When complete, the track matrix appears as shown in Fig 8.

As has already been mentioned, the system is not a simulator in the normal sense; it will intervene to prevent the student following a strategy that is at variance with the instructor's wishes. In order to maintain synchronisation between the apparent real time sections and the prompt and feedback branching required, the basic unit of the scenario build is the **event**. An event was defined as an instant in game time. An event can be anything an instructor wishes; a text prompt, a track, a tote screen. Having designated the sequence of events, the instructor inputs these requirements with their associated data i.e. text data or track parameters into the scenario build matrix Fig 9. The toolbox is designed to allow tracks to divide, with a specified maximum of total tracks on the screen at any one time.

In order to allow the instructor to review, modify and test scenarios the following features were built into the toolbox design:

- Scenario Play
  - Normal speed, forward and back
  - Fast speed, forward and back
  - Freeze
- Scenario Edit
  - Addition or subtraction of tracks or radar contact
  - Prompts - text or audio can be added at any point
- Scenario Store
  - Store to a library for use or edit

The completed scenario in Fig 9 appears as to the student as depicted in Fig 10.

SOFTWARE ISSUES

As detailed modelling of the ADAWS system itself is not required, the software engineering issues seem at first glance to be trivial. This is far from true. This software lies somewhere between an accurately modelled simulation and a more traditional CBT solution. It is vital that the software is fully specified and the design work is comprehensive. In short, the text book approach to software design is appropriate in this case. This is not always the case for CBT ! Of crucial importance to this project is the handling of data. Although the data appears to be the property of the instructor, it, or parts of it, must also be available to the student. The student requires the facility to add or amend data, dependent upon the situation. This requirement is made more complex by virtue of the fact that these data transfers must appear to take place in real-time. It is therefore vital that data flows are accurately specified, and appropriate data structures are employed. Related to these data issues are factors such as:

- Disk access time

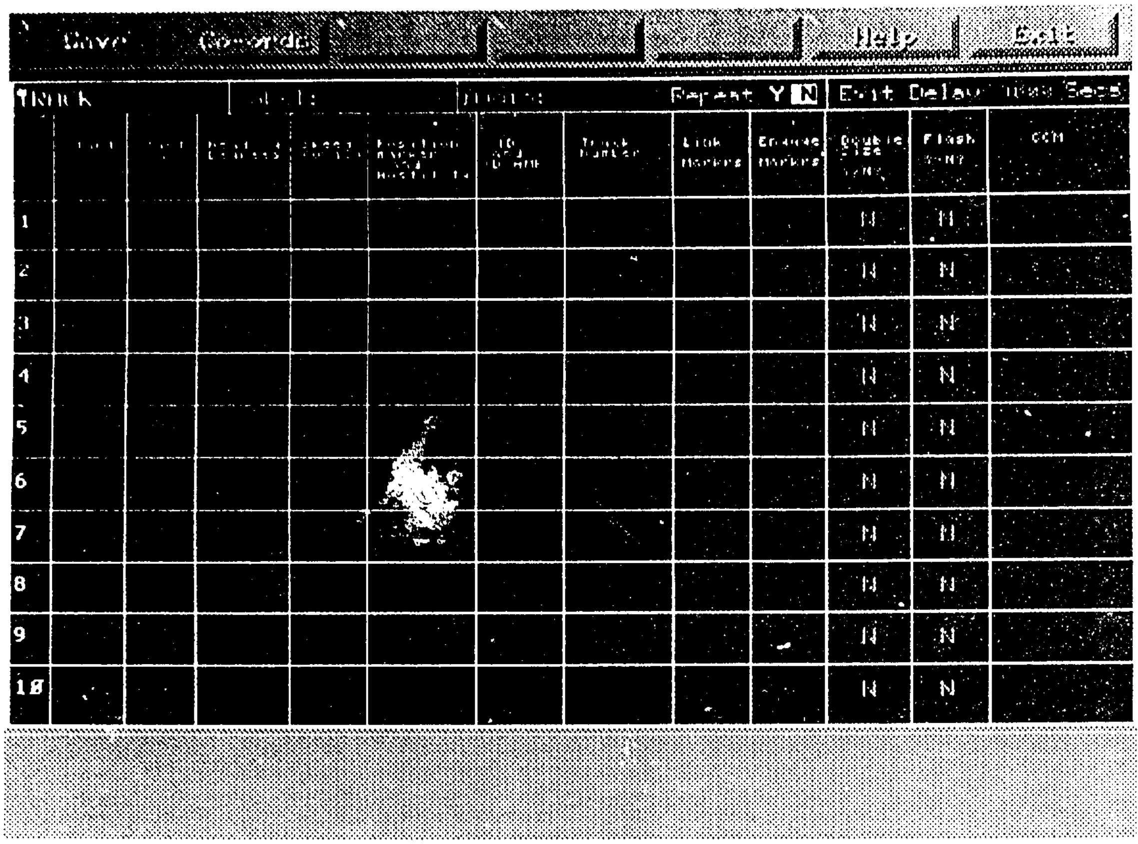


Figure 7 Track Data Matrix Shell

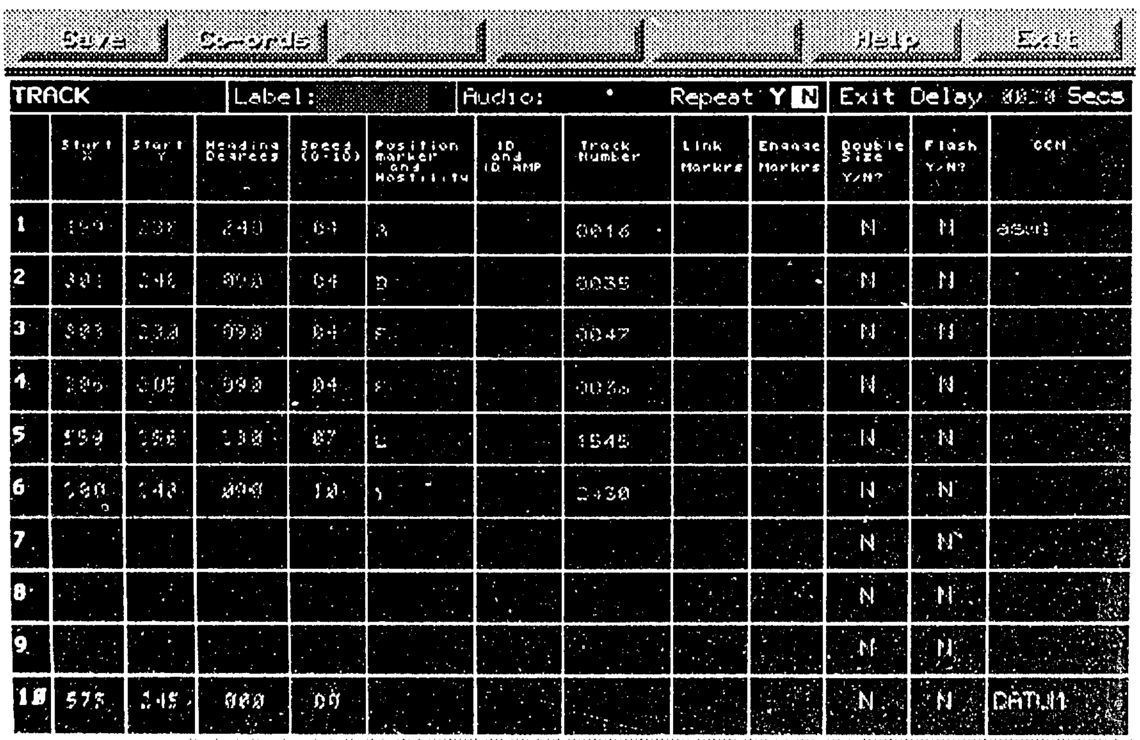


Figure 8 Completed Track Data Matrix

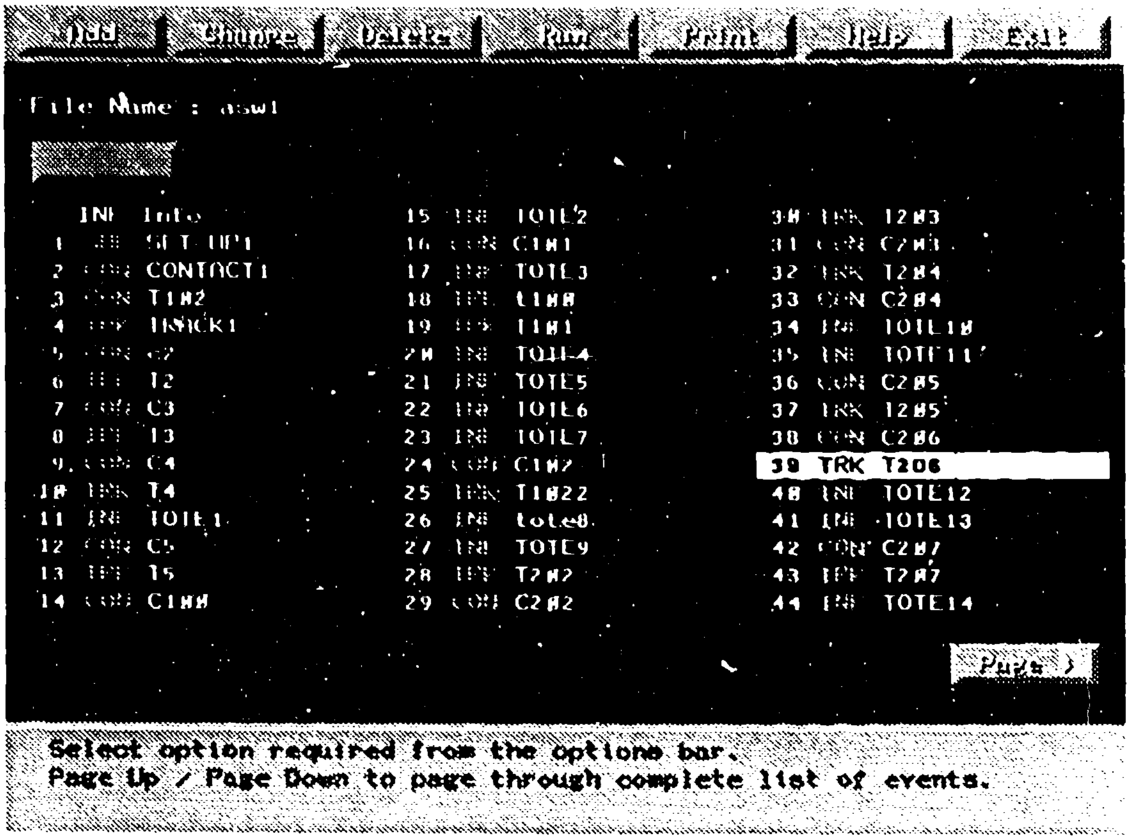


Figure 9 Scenario Event Builder

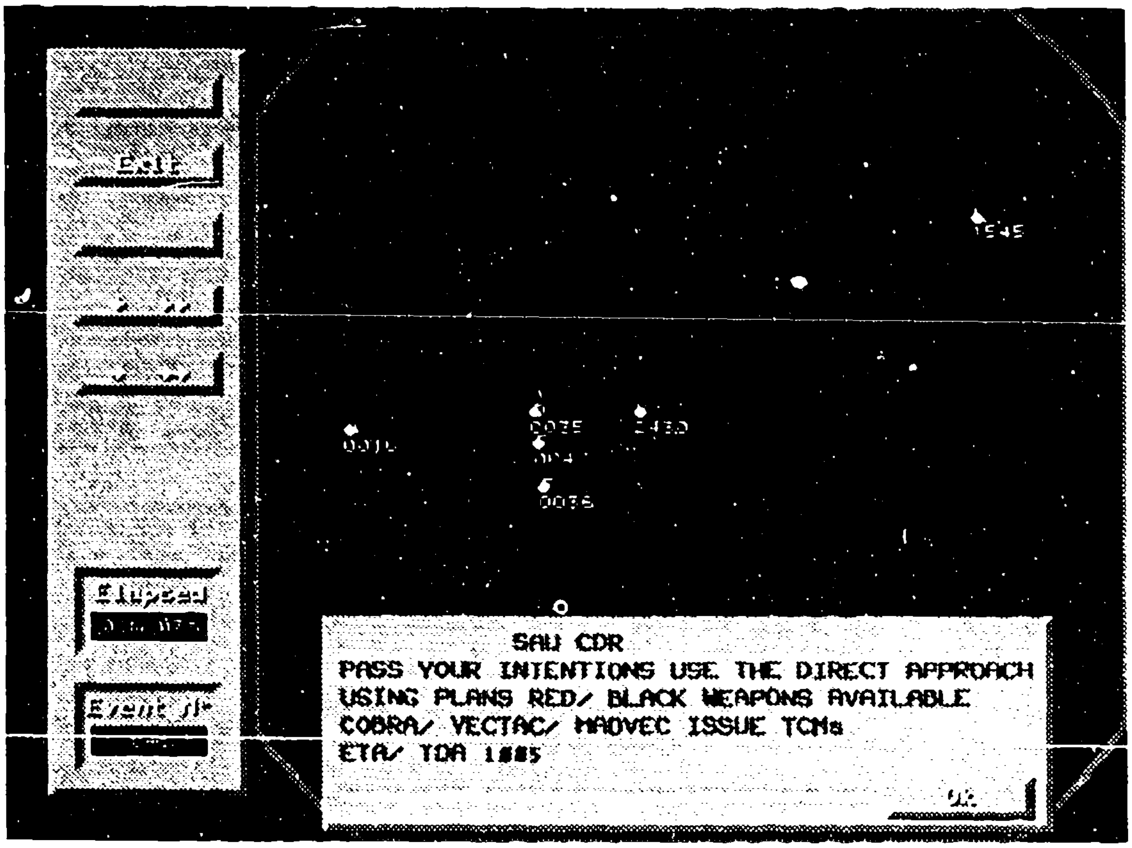


Figure 10 Student Display Screen

- Complexity of audiofiles
- Memory requirement for data stored in memory
- Shared memory
- Data update mechanism

Another main issue that requires careful consideration, relates to the use of the non standard input devices. Clearly device drivers must be written, but if the response of the keyboard is to closely replicate the behaviour of the actual system, considerable design effort must be invested in matching the keyboard inputs to the computer's BIOS.

ASSESSING SUCCESS

At the time of writing a detailed analysis of the training effectiveness of the ADAWS courseware is not available. The operational equipment is only just entering service and the training throughput has been restricted to the crew of the first ship to receive this fit. The fact that it has been possible to provide a degree of procedural training ahead of the actual equipment entering service is in itself something of a triumph for the design concept.

Currently however, the most tangible evidence for the success of this courseware is the initiatives that have developed from the original concept. Firstly, the scenario generation feature has been used to create source materials for voice procedures training (RTQ). The RTQ facility has replaced an ageing language laboratory type trainer that used relatively primitive representations of tactical situations for the students to report upon. Now at little extra cost, the facility has been improved to provide realistic and dynamic situations for training. The instructors have found it possible to introduce elements of team training into their curriculum which hitherto, was not feasible outside of the main operations room simulator. The second initiative to come from the project is a requirement to provide a replacement training facility for the original ADAWS CBT facility. Since the toolbox is essentially generic, provision of a dedicated keyboard and additional libraries enables the system to function as either Mod 1 or the original Mod 0. The flexibility of resources and the courseware productivity gains make the business case easy to establish.

Perhaps the most significant development to stem from the ADAWS concept is a toolbox CBT requirement for the RN's latest command and control system SSCS, due to enter service in the Type 23 frigate. The underlying training concept being planned is identical to that described for ADAWS but this system will reflect the increased sophistication of SSCS. The system will encompass both surface and sub surface pictures and will feature the multifunction console concept and reconfigurable plasma panel keyboard using a touch screen panel for its primary input device.

CONCLUSION

The RN recognises that instructor involvement is a vital ingredient of effective CBT solutions. Early attempts at actually using instructors to produce materials were only partially successful and it was apparent that the resources needed to implement this approach could not be sustained. A solution investigated by the RN was the creation of software toolbox facilities, designed to meet clearly identified training needs. These proved both economic and effective.

The concept was extended to incorporate facsimile peripherals in order to increase realism. This extended the use of CBT to much higher levels of training objective than had previously been possible. The approach was pioneered for the upgrade to the RN's ADAWS system ADAWS Mod 1, and this has enabled practical skills training for crew members prior to them joining the first ship to receive the upgrade. It has been adapted for voice procedures training and proposals are in hand to extend the training to ADAWS Mod 0. Design specification work is in hand to develop a similar system the RN's latest command and control system SSCS.

REFERENCES

1. "CBT in the Royal Navy - Training the Trainers". PENGELLY, MOORE & DOWSETT, I/ITSEC, 1992
2. "Computers in Training - A Guide for Trainers", sponsored by the [UK] Joint Standing Conference on Training Support (JSCOTS), 1989.
3. "Computer Based Training - Hit or Myth?" MOD (DNETS), I/ITSC, 1988.
4. "Guide to the Use and Procurement of CBT" Royal Navy CINCNAVHOME, Oct 1990.
5. "Simulation and toolboxes for Sonar Skills Training". J. HALL., ITEC 1990.
6. "Producing Instructional System". Romiszowski A J., Kogan Page, 1984.
7. "Domain Knowledge Representation for Instructional Analysis". JONES M.K., Li Z., Merrill M.D., Educational Technology, 1990, 30 (10).
8. "A Workstation Based Integrated Training System and its Application to Electronic Warfare Training". Knight & Morgan, ITEC, 1991.

HUMAN-COMPUTER INTERFACE AND TRAINING ISSUES IN THE DESIGN OF AN EXPEDIENT COMPUTER-BASED LANGUAGE TRAINING PROGRAM

Daniel E. Mullally Jr., J. Peter Kincaid and George Kishek
Institute for Simulation and Training, University of Central Florida
Orlando, Florida

ABSTRACT

This paper describes the development and user concept evaluation of a Computer-Based Language Training Program in Somali titled "Humanitarian Expedient Language Pronunciation Simulation" or HELPS. The HELPS Concept Demonstration Project was designed to provide expedient language training to Marines involved in humanitarian relief duties in Somalia during Operation "Restore Hope". The HELPS project was a joint cooperative effort by the Institute for Simulation and Training (IST) at the University of Central Florida, who donated the software program without cost to the Marine Corps, Apple Computer, Inc. who loaned the 10 Macintosh® PowerBook™ computers to the Marine Corps for the duration of the project, and the Marines of all ranks in I MEF who evaluated the HELPS Concept Demonstration Project in Somalia.

The analysis, design, and development steps of the HELPS project are outlined. These steps allowed the rapid prototyping and delivery of HELPS to Marines deployed in Somalia in seven weeks from concept to delivery. The results of the user evaluation in Somalia is analyzed and presented.

This paper has several objectives. The first is to describe the analysis, design, and development of the HELPS project. The second is to describe the unique human-computer interface issues involved in the design of the HELPS. The third is to present the results of the user evaluation and acceptance of an expedient language training system. The fourth is to demonstrate and summarize the implications of a capability to bridge the language barrier in computer-based language training.

ABOUT THE AUTHORS

Daniel E. Mullally Jr. is a Research Associate at IST. He has 20 years of military experience as a Marine Corps Officer followed by over 13 years of training development experience. He has served as a training consultant in industry, a scenario designer in manual and computer-based military simulations (including the 14th and 15th I/ITSECs Distributed Interactive Simulation Interoperability Demonstrations), and a subject matter expert in military training. His research activities include the design and development of a Forms Translator Assistant for the U.S. Customs Service using computer-assisted text and speech presentation methodology and the USMC HELPS Project.

Dr. J. Peter Kincaid is a Senior Scientist with IST. He is a human factors scientist with over 20 years of research experience working in human factors, simulation and training for each of the military services, combined with more than 15 years of university teaching experience. From 1979-1985 he was one of the Navy's lead scientists for English as a second language training courses, working with the Defense Language Institute. He developed the DoD readability standard, the Kincaid Readability Formula. He has served as a NATO lecturer, and as a peer reviewer for the Office of Naval Research, the Air Force Human Resources Laboratory, and the Army Research Institute and is listed in American Men and Women of Science.

George A. Kishek is employed by the NCR/AT&T corporation as a Systems Engineer. He was the principal software programmer for the HELPS Project in the Computer Assisted Language Lab. He is currently working towards a Doctorate in Computer Science at The University of Central Florida.

HUMAN-COMPUTER INTERFACE AND TRAINING ISSUES IN THE DESIGN OF AN EXPEDIENT COMPUTER-BASED LANGUAGE TRAINING PROGRAM

Daniel E. Mullally Jr., J. Peter Kincaid and George Kishek
Institute for Simulation and Training, University of Central Florida
Orlando, Florida

INTRODUCTION

HELPS Concept Development

Project HELPS (Humanitarian Expedient Language Pronunciation Simulation) was created at the Institute for Simulation and Training (IST), as a Concept Demonstration Project. This HELPS concept had originally been conceived and demonstrated as an application of several foreign language research and development projects conducted in the Computer Assisted Language Lab (CALL) at IST. These projects included "PEDRO", a computer-based "English as a Second Language" (ESOL) project, the U.S. Customs Service Forms Translator Assistant (FTA)[1] project, and "SURVIVE", an early demonstration program of the HELPS concept. Previous research into language training using computers with a voice interface revealed a high order of training transfer and effectiveness.[2,3]

Conflicts around the world and the need to intervene created a pressing need for U.S. marines to be able to communicate in a rudimentary way with the indigenous population. Areas of needed functional communication skills include both military and humanitarian operations. The need for an efficient, cost effective and rapid means to learn new language phrases to effectively interact with others will not end with Somalia. The HELPS concept is ideally suited for rapid development of a limited vocabulary and a set of functional phrases specifically geared to provide military personnel sufficient skills to communicate with indigenous personnel. In February of 1993, General Carl E. Mundy Jr., USMC, the Commandant of the Marine Corps, stated that "Americans will collaborate increasingly in global efforts at conflict resolution, humanitarian and disaster relief, and nation-building in places just like Somalia."[4]

Currently, nearly all newly enlisted Marines are high school graduates but only 18% have achieved any proficiency in a foreign language. Foreign language instruction in U.S. public schools leaves much to be desired; the emphasis is placed on reading and grammar rather than speaking and listening.

The traditional model for learning to speak a foreign language is for the instructor to verbalize words and phrases and have the class respond as a group. The instructor tries to hear incorrect pronunciations and provide individual feedback. The high ratio of students to teacher leads to slow, inefficient learning. Many language classes use laboratories in which students listen to tapes and repeat the phrase into microphones. The instructor can monitor student responses one at a time and provide feedback. Assuming a student to instructor ratio of 20/1, students receive individual feedback only on an average of 5% of the time. In addition, it is awkward for the student to hear a recording of his response for comparison with the correct one. Computer-Assisted Language Learning (CALL) is a relatively recent technology which shows promise in improving the traditional model.

The underlying concept of the HELPS project was to demonstrate the technology of using a computer to:

- present foreign language (Somali) and English text on screen,
- recall recorded native speech,
- record the user's attempt to mimic the native speaker,
- allow the subjective comparison by the user of his/her recorded voice against that of the recorded native speaker, and,
- provide repeated interactive cycles of the listen/record/compare process as a means to quickly learn new foreign language words and phrases.

The HELPS technology approach, the Human-Computer Interface issues, the

hardware selected, the software design approach selected, and the personnel assets available were critical decisions faced in the initial stages of the HELPS concept development Project.

This technology demonstration was designed to fill an immediate need to train Marines engaged in Humanitarian operations in Somalia during Operation "RESTORE HOPE" in the rudiments of speaking Somali. The timely delivery of an adequate language training device was of overwhelming importance and dictated several design and development innovations in the creation of the HELPS Project.

Hardware Availability - Hardware for the HELPS Concept Demonstration Project was limited to those readily available, off-the-shelf computers which had the ability to record and playback speech files. The computer selected for the Concept Demonstration had to have a capability to use headphones to diminish noise pollution, create a distraction for others nearby, or draw attention if used in a combat situation. The computer selected had to have the ability to be used in transit in ships, aircraft, or vehicles as required. The computer had to be capable of portable operations for extended periods of up to one hour. The computer had to have the ability to use a variety of field or expeditionary power sources. The delivery platform had to be rugged, lightweight and easily serviced. The computer, to be fully capable of meeting the expected range of field use, would require the capability to provide sound output to an external amplifier, bullhorn or loudspeaker. The computer used in HELPS would require a high order of user-friendliness for the first-time user who would possess a minimum of computer training or keyboard familiarity.

The PowerBooks™ used in the development of the HELPS system were loaned to the Marine Corps for a period of six months specifically for use in this HELPS Concept Demonstration Project.

Software Selection - The software selected for the HELPS Concept Demonstration Project was required to be readily available, inexpensive, and user friendly. The speech files and text supported by the software had to be rapidly accessed, and provide for a "FIND" function by searching for any portion of the word or phrase sought. The software had to be compatible with the delivery of sound files on the hardware selected for the HELPS Concept Demonstration Project. The software used in the HELPS Concept Demonstration Project was provided without charge to the United States Marine Corps forces engaged in humanitarian relief operations in Operation "RESTORE HOPE."

Personnel Assets Available - Personnel, with the variety of specialized skills and talents needed, had to be readily available from within the University of Central Florida or in the immediate Orlando, Florida area in order to provide HELPS as rapidly as possible to the Marines in Somalia. The availability of computer software, voice recording, English language specialists, military task analysts and Somali language translators had to be quickly determined in advance of the development of the HELPS project. Fortunately all of the requisite personnel were available from the University community. A skilled Somali translator, Abdi-Rizak I. Salah was found through the International Student registry. Abdi proved to have a vast knowledge of the language, culture, and conditions that prevailed within his homeland, and an ability to select and tailor the words and phrases chosen for the HELPS Concept Demonstration Project. The USMC provided initial and follow-on HELPS Concept Demonstration Project evaluation personnel assigned to the I MEF Headquarters in Camp Pendelton, California and in Somalia.

Objectives of the Concept Demonstration

The objectives of the HELPS Concept Demonstration Project were to: (1) take advantage of recent advances in computer technology which support language training; (2) design a <u>functional</u> instruction course specifically geared to teaching "survival" Somali speaking skills for particular military missions in as short a time as possible; and (3) produce and test this courseware for proof-of-concept within six weeks (or sooner) after program start.

The HELPS Concept Demonstration Project in Somali objectives were structured for FMF units preparing for deployment, embarked on ships for extended operations offshore, or currently ashore in Somalia during Operation "RESTORE HOPE". The HELPS Computers would be placed within units for any of these situations, and made available to individuals on a scheduled, 24 hour basis. The placement of

computer-based expedient language trainers within their units would provide a learning center dimension for Marines and sailors to develop a language skill critical to any meaningful interaction with the Somali speaking people during humanitarian operations.

Rapid Prototyping and Development - The HELPS Concept Demonstration Project was designed to rapidly prototype and develop a system in response TO a perceived need to quickly provide a system for evaluation under the initial conditions of the intervention. The seven week period of development forced some shortcuts in design and content completeness in favor of early delivery. That rapid prototyping and development experience is documented in this report to assist others in future projects of this type.

Six Month Evaluation Duration - The HELPS Concept Demonstration Project was designed to evaluate the performance of the system during a six month period. The underlying rationale of the demonstration was to provide a user evaluation under field conditions of a computer-based language training system. The evaluation was not meant to examine in depth the content or effectiveness of the phrases selected for inclusion of the system, but rather it was meant to determine the level of user interest in continued development of a future system. Without some statement of user acceptance the continued development of a system could not be advised.

Technology Push - The HELPS Concept Demonstration Project was designed to provide a window of opportunity to the ultimate users of the proposed system (Marines in Somalia engaged in Operation "RESTORE HOPE") of the modern state-of-the-art computer technology available to provide them with a Computer Aided Language Learning capability.

ANALYSIS

User Requirements Analysis

Analysis and design of the user requirements proceeded from previous experience with "SURVIVE" and "PEDRO" computer based Spanish language programs developed at IST. The selection of the hardware, software, and personnel assets, and the time available structured the analysis and design process. The initial step in the HELPS analysis was to select the appropriate language content to teach. The language requirement is more for speaking than for reading. The expedient language training should be computer-based, self-paced and designed to be delivered efficiently aboard ship, other work places, or in the field.

If the purpose of HELPS was to provide expedient language training as a means to communicate with Somali natives, then the words and phrases selected for inclusion in HELPS needed to be complete communications. Communications is a process which is transactional in nature. Communications to be complete needs feedback to the speaker from the listener.

Consideration was given to the selection of the native Somali speaker. The issues of dialect and regional inflections were overcome by selecting a native Somali speaker born and raised in Mogadishu, the capital of Somali. Mogadishu is considered the cultural center of Somalia and due to the influence of radio broadcasting from Mogadishu is viewed as the predominant accent heard and understood throughout Somalia.

Written material was provided to support the computer-based language instruction in the form of a vest-pocket sized booklet appropriate for self-study. (See Figure 1). The courseware was designed to be so easy to use as to encourage intensive practice. The phrases were carefully scripted in a format designed to provide on separate lines a numbered English, a normal Somali, and a phonetically structured Somali Phrase. This phonetic Somali phrasing took the convention of a hyphenated presentation with the accented syllable being capitalized to show the emphasis. The presentation and numbering conventions were standardized for both the computer screens and the vest-pocket sized phrase book.

The configuration of the computer-based delivery device was designed to provide the user with a friendly, intuitive interface offering a book-like presentation of separate pages for each phrase. The turning of the page was facilitated by using the directional arrows at the bottom of each screen which allowed the user to "page" backwards and forwards to other individual phrase screens.

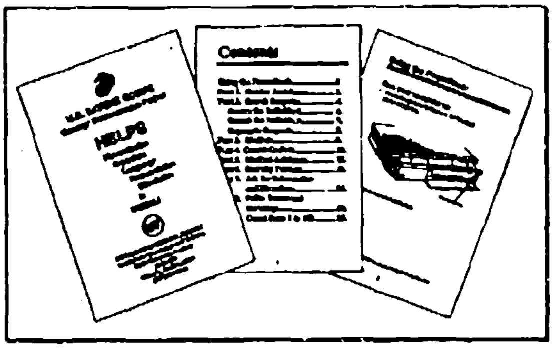


Figure 1. HELPS Instruction Booklet

Phrases were selected to meet the conditions expected of a military unit engaged in both military and humanitarian operations. Some of the phrases were chosen from Marine Corps training manuals to allow the Marine Corps training manuals to allow the Marines to select from categories of Search, Secure and Segregate individuals.[5] Courseware was designed to be supplemented by audio-tapes.

Criteria for Word and Phrase Selection

Previous experience in testing IST developed language training projects lead the analysts to select those statements which are imperative in their nature and require only compliance to determine if communication is complete. Such statements as "STOP" or "COME HERE" are considered completed communications if the listener or listeners respond as requested.

The second type of communications selected for inclusion in the HELPS Project is the phrase carefully designed to elicit a non-verbal response. Phrases like "SHOW ME WHERE IT HURTS" or "HOW MANY ARE WITH YOU?" are examples of these phrases. The listener completes the communication by pointing or by responding with a show of fingers to these questions.

Human Computer Interface Issues

The limitations imposed by the computer-based language training approach were examined in detail to determine the natural process of listening, repeating, mimicking, correcting, and listening again and again as required to capture the example of correct pronunciation provided by the traditional language tutor. The HELPS user can accomplish all of these steps in language learning except to receive the acceptance or rejection of his/her pronunciation attempt. That immediate feedback, oral or non-verbal, provided by the traditional language tutor is replaced in HELPS with the subjective personal evaluation of the user's recorded performance by the user himself/herself.

This departure from the traditional language tutoring process in the unique Human Computer Interface design of HELPS offers some interesting observations. First, the user can hear his/her own recorded voice compared against that of the native speaker. This ability to hear their own electronically reproduced voice effectively places the user "outside themselves". The user is offered an accurate representation of their own voice and the expert native speaker's voice to compare without any implied sense of embarrassment at their inexpert repetition. Without the fear of failing to perform to standard, the user can repeat the listening, recording, and comparing cycles offered in HELPS as many times as they feels it is necessary.

This is the reason the word **SIMULATION** was chosen for inclusion in the HELPS title. The computer acts as a simulated tutor. The user can demand and receive the continued, uninterrupted, and private performance of the tutor (HELPS) to meet his/her personal training objectives.

DESIGN

The HELPS project grew from previous research in Spanish and Arabic computer-based language training courseware developed at the Institute for Simulation and Training at the University of Central Florida. HELPS was designed to be based on the Apple Macintosh® PowerBook™ to allow the user to record his/her own voice and compare it with the recorded speech files of a native speaker. The student is tutored by a native speaker (whose voice resides on the computer) taking advantage of both auditory and visual feedback.

Intuitive "Point and Click" and on-screen instructions coupled with a pocket-sized phrase book provide the user with a rapid entry into the HELPS system. The use of icons depicting the intended functionality were incorporated throughout the screen design. These icons were explained in detail in the HELPS phrase book introduction.

The PowerBook™ was set up to open automatically to the HELPS program. This allowed the user to see the first opening screen immediately upon start-up. (See Figure 2)

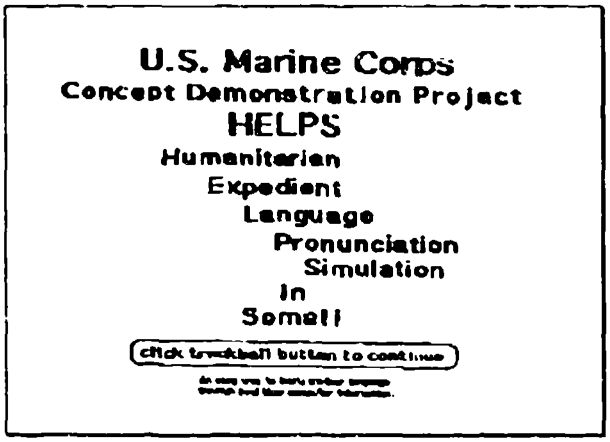


Figure 2. HELPS Opening Screen

Subsequent screens are reached by "Point and Click" techniques. To keep with the familiar "Book" metaphor the first operative screen contains the "Table of Contents" offering any of 10 lines active to the trackball and trackball buttons "pointing and clicking" technique. (See Figure 3)

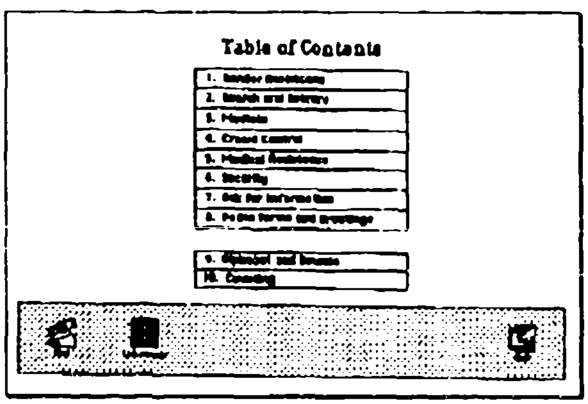


Figure 3. HELPS "Table of Contents" Screen

To quickly reach an appropriate phrase or word a "Find" function was provided which can instantly present the choices for selection. After selecting a subject from the "Table of Contents", the user is presented with a lined screen displaying a choice in English of up to fifteen words or phrases. By pointing and clicking on any one of these lines the user is presented with the selected word or phrase screen. The expedient language instruction is presented on a segmented screen with the Somali word or phrase shown boxed first in a "normal" Roman alphabet and secondly, boxed in a hyphenated format showing the accented syllables in all upper case letters. The English word or phrase is shown in a box below the Somali translations. Both the "normal" speech and the "phonetic" speech boxes have active sound files. Clicking on either of the Somali boxes, the user is presented with the recorded phrase spoken by a native speaker. (See Figure 4)

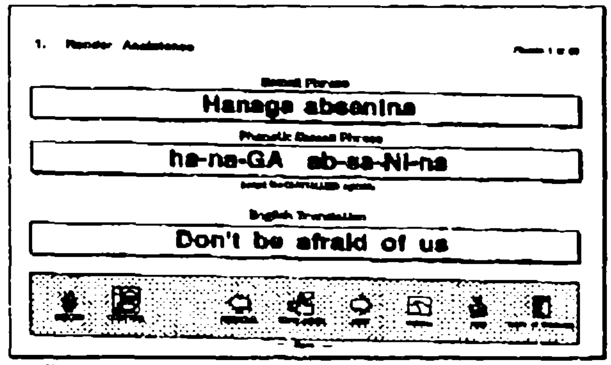


Figure 4. HELPS Sample Phrase Screen

A unique feature of the HELPS software is its ability to link up to fifteen selected words or phrases. The user can create and save these LINKED screens to replay upon demand. He/she can modify the previously linked screens by deleting or adding new phrases (See Figure 5). The phrases are separated by a built-in pause simulating the pause of a speaker at the end of each sentence or word. The resultant phrases can be delivered directly from the PowerBook™ or delivered to an external speaker.

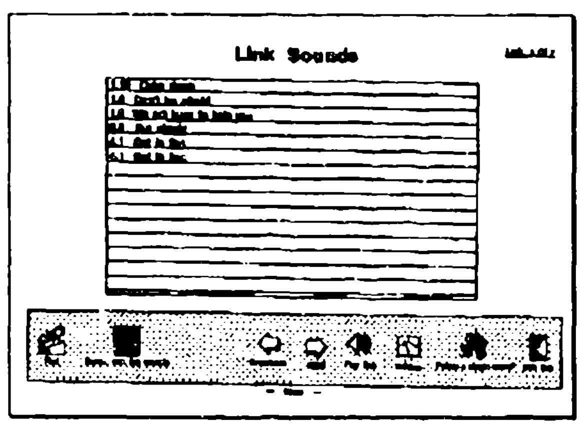


Figure 5. HELPS "Link" Screen

In addition to each of 253 words and phrases, the 21 consonants are shown in combination with each of the 5 possible vowel

and 5 double vowel and the 32 examples of numbers contained in the counting section are presented. The HELPS system allows for the repeated self-paced attempts to master difficult phrasing and pronunciation until the user is satisfied with the results.

DEVELOPMENT

Initial Prototype Test

The initial prototype tests were conducted at the Institute for Simulation and Training at the University of Central Florida using sequentially, the HELPS Concept Demonstration Project developmental personnel, other students from within the University, Marines from the Naval Training Systems Center, and Marine reservists from the 4th Truck Company in Orlando, Florida. These tests provided insight and feedback to the design deficiencies of the initial prototype.

Problems Encountered

The initial prototype development item was modified and transported to Camp Pendelton, CA. Evaluation and modification of the HELPS System continued concurrently during the demonstrations and initial training for USMC Project personnel. Several problems were encountered with the design and development of the speech files during this period. These problems were analyzed and modifications to the system were made as required. Software modificationas were made to allow the user a period of time to record their voice timed against either the normal or slow paced Somali presentations.

Initial Prototype Distribution

The initial prototype system, modified as the result of user comments in the initial prototype tests at the Institute for Simulation and Training, was returned to the Marine Corps on 25 January 1993 at Camp Pendelton, California. Six of the systems were transported into Somalia by the Marine Corps Project Officer, and two systems were retained for training use by the Marine Expeditionary Unit (MEU) at Camp Pendelton, scheduled to replace the MEU in Somalia. One System was retained by the Institute for Simulation and Training and one system was retained at I MEF Headquarters at Camp Pendelton.

USER EVALUATION RESULTS

Summary of HELPS use in Somalia

Preliminary analysis of results resulted in the following conclusions Testing provided a successful proof-of-concept. Every subject tested had an overall positive response to the HELPS. The user evaluation responses to the HELPS project are being collected and evaluated at the time of this writing. A copy of the Evaluation Questionnaire Form used to elicit responses from HELPS users in Somalia is attached to this paper. Several initial revisions to the human computer interface were made as a result of responses to this evaluation form. Suggestions for improvements included repeated requests for additional words and phrases. All of those responding agreed on the requirement of having the HELPS system available for training in advance of operations.

The HELPS is cleany a viable concept which has been shown to achieve its design objectives.. Preliminary findings in field tests indicate a high degree of user acceptance. The intuitive design features of the HELPS interface lead to a rapid learning curve in achieving operator understanding. The HELPS system can defuse a portion of the anxiety caused by entering a new language environment by providing personalized, interactive instruction with the rate of delivery determined by the user.

IMPLICATIONS OF CONCEPT DEMONSTRATION

Technology Transfer

The HELPS approach, based on continued promising research and development at IST, provides for the individual tutoring of a student learning to pronounce words, phrases and sentences in any foreign language. Training time is predicted to take 5-10 hours of time at the computer (plus additional self-study using audio-tapes and a booklet) to learn 50 functional phrases.

Other HELPS modules using the same approach and authoring system can be developed for learning any foreign language. Programs could provide for a variety of help functions including the closest English approximation to the selected sound. A "see and hear" component of the help function, which could also be implemented, would depict: (1) a

"talking head" animation pronouncing the sound in question and providing advice on how to produce the sound; and/or (2) wave forms of the subject's and native speaker's production of the sound presented for easy comparison. IST is also evaluating the feasibility of incorporating voice recognition to provide feedback on correctness of pronunciation.

Future language programs will take advantage of emerging micro-miniaturization to develop language training programs on "Palm-Sized" computers. The introduction of this technology offers the ability to create a truly individual language training and communications device. Talking technical manuals are easily conceivable.

REFERENCES

1. Daniel E. Mullally Jr. "U.S. Customs Service Forms Translator Assistant (FTA), Final Report" IST-TR-92-08, Institute for Simulation and Training, University of Central Florida, April, 1992

2. Kincaid, J. P., Mullally, D., & Kincaid, J. F. "Language Training Using Computers With Voice Interface." Educational Technology, January, 1992.

3. Kincaid, J. P., Thomas, M., Lindeis, A., & Arnold, D. "Use of Voice Technology for Language Training." Proceedings of the Annual Meeting of the Human Factors Society, September, 1992.

4. General Carl E. Mundy Jr., USMC, Commandant of the Marine Corps "The U S Marines Are Old Hands At Humanitarian Intervention." Armed Forces Journal International, February 1993.

5. USMC Individual Training Standards (ITS) System, Occupational Field (03) Infantry, Battle Drill Guide, U. S. Government Printing Office, 1988 p.3-4-87-89.

A Cognitive Science Approach to Structuring Lesson Content

ABSTRACT

Patsy D. Moskal and Patrick J. Moskal
Institute for Simulation and Training
University of Central Florida

Robert H. Ahlers
Naval Training Systems Center

A critical component of combat readiness lies in the skills and knowledge of the deployed personnel. However, these skills are highly perishable without continued training. Embedded training (ET) is one potential solution to the problem of maintaining a maximum level of operator readiness. The objectives of ET are to build on existing knowledge, diagnose and correct deficiencies as efficiently as possible, consolidate skills through practice, and acquire new knowledge and skills. ET effectiveness can be increased by implementing instructional technologies that promote efficient acquisition and retention of skills and knowledge. Current research on the application of cognitive learning principles to training provides precise instructional methods and implementation techniques. Recent research at the Institute for Simulation and Training (IST), in collaboration with the Naval Training Systems Center (NTSC), has demonstrated the power of this cognitive learning approach in applied Navy training environments. Significant improvements were found in the instructional capabilities of tactical console ET lessons.

The present effort involves evaluating this instructional methodology using a Computer-Aided Submode Training (CAST) lesson of the Navy's Aegis weapons system. CAST was selected because it provides an ideal environment for implementing cognitively-based instructional enhancements. It incorporates a well-developed ISD methodology, which provides a framework to build a more specific cognitive learning approach. A CAST lesson was restructured according to the cognitive task analysis methodology. Performance on the cognitive lesson was then empirically compared to performance achieved on the original lesson. Trainees receiving the cognitively structured lesson significantly outperformed trainees receiving the original lesson by an average improvement of 47%. These findings strongly support previous research concerning the merits of this cognitive approach to learning.

About the Authors

Ms. Patsy Moskal is a Research Computer Scientist at IST. Her research involves developing instructional systems for education and industry which combine intelligent tutoring system technology, expert teacher instructional strategies, and the use of graphics, animated simulation, and hypermedia. She is also involved with research on applying cognitive learning principles to operational Navy combat systems. She holds an M.S. in Computer Science from the University of Central Florida and is currently working on an Ed.D. in Curriculum and Instruction from the University of Central Florida.

Dr. Patrick Moskal is a Research Psychologist with the Institute for Simulation and Training. His current research interests include intelligent tutoring technology implementation and evaluation, performance measurement and performance under stress, and simulator fidelity determination. He holds a Ph.D. in Experimental Psychology from the University of Notre Dame, specializing in sensation and perception, and psychophysical scaling. He has numerous publications and presentations related to his work at IST, and previously at NTSC.

Dr. Robert Ahlers is a Research Psychologist with the Human/Systems Integration Division of NTSC. He has managed research projects concerned with the application of knowledge-based modeling to the simulation of intelligent agents within a training environment. His recent work addresses the use of artificial intelligence techniques to provide certain instructional features, such as performance diagnosis and feedback, instruction sequencing, and tactical adversary capabilities, within an ET environment. He graduated from the University of Virginia with B.A. and M.A. degrees in Experimental Psychology and from North Carolina State with a Ph.D. in Human Factors.

A COGNITIVE SCIENCE APPROACH TO STRUCTURING LESSON CONTENT

Patsy D. Moskal
Patrick J. Moskal
University of Central Florida
Institute for Simulation and Training
Orlando, Florida

Robert H. Ahlers
Naval Training Systems Center
Orlando, Florida

INTRODUCTION

A critical component of combat readiness involves the skill and knowledge levels of the personnel who must perform the specified missions. For Naval readiness, key personnel must competently operate sophisticated weapons, electronic warfare, navigational, and communications systems. In order to skillfully operate these complex systems, operators must be well trained initially, and they must receive frequent opportunities to practice and refine their skills. The cognitive and psychomotor skills needed to operate these devices are highly perishable, and without continued refresher training they will degrade rapidly (e.g., Massey, Harris, Downes-Martin, & Kurland, 1986).

On-board embedded training (ET) is one potential solution to the problem of maintaining a maximum level of operator performance and readiness. By building training capabilities into, or adding them onto, an operational system, operator skill proficiency can be maintained and enhanced in an accessible, high fidelity environment. In order to maximize training efficiency and retention, lessons should be structured to provide each trainee with the most learning gain from each training session. The goal of the ET session is to build on existing knowledge, to diagnose and correct deficiencies as efficiently as possible, and to allow for consolidation of skills through practice. The training effectiveness of ET sessions will be increased with the implementation of instructional technologies and strategies that promote efficient acquisition and retention of skills and knowledge.

Substantial cognitive science research has been directed at understanding the cognitive processes involved in knowledge acquisition and skill development. Learning to perform complex tasks, such as operating a tactical console, involves an active knowledge construction process, and this construction of new knowledge is highly dependent upon both existing knowledge and the situation or environment in which the learning takes place (e.g., Williams, Reynolds, Carolan, Anglin, & Shrestha, 1989). The cognitive model formed by this knowledge acquisition process is used in future interaction with the device. Cognitive science research has indicated that the underlying structure of this model is consistent with the framework of a production system (Anderson, 1983, 1986). A production system consists of a data base of facts, or declarative knowledge, and a set of productions, which are IF-THEN rules. This implies that if we develop instructional material consistent with a rule-based, production system approach, learning should be aided and improved, and the application of that learned knowledge will be facilitated.

Considerable research has been focused on the notion of a rule-based approach to learning. Nisbett and associates (Nisbett and Kunda, 1985; Cheng, Holyoak, Nisbett, and Oliver, 1986; Fong, Krantz, and Nisbett, 1986) performed a series of experiments which demonstrated that people are very good at abstracting rules from examples, as long as they can relate to those examples. Students were more likely to generalize and apply formal rules of logic to everyday problems when given

pragmatic examples from which to abstract the rules.

This rule-based approach to knowledge acquisition also emphasizes the importance of goals in a hierarchy of rules. The importance of goals in information processing can be seen in production models of cognition (Newell and Simon, 1972; Anderson, 1983; Holland, Holyoak, Nisbett, and Thagard, 1986; Klahr, Langley, and Neches, 1987). These models also address the importance of a hierarchical goal structure which guides processing. One important function of a goal is in directing attention. Research has also demonstrated that goals can set the speed of successful perceptions as well as what is perceived (La Berge, 1973; Posner and Snyder, 1975). In addition to the importance of goals in guiding attention and actions, a hierarchy of goals also plays a prominent role in memory organizations. Anderson (1983) has demonstrated the importance of a hierarchy of goals in guiding the development of rules (productions) and the creation of new rules by combining existing rules. Jeffries, Turner, Polson, and Atwood (1981) and Anderson (1986) have noted that by establishing a goal tree, the rules that are likely to be composed by students can be predicted.

The cognitive science research effort has led to the formalization of various learning processes and strategies. The application of research in cognitive skill acquisition has demonstrated the feasibility of the approach and its positive impact on the learning process (Anderson, 1983, 1986). For example, Chi and associates (1981, 1989) showed that good students are able to explain the conditions associated with a specific action, whereas poor students are not. Good students also generate more complete condition-action rules during the learning process. They construct such rules from the instructional material presented to them (Bovair and Kieras, 1990). Research demonstrates that learning is enhanced when material to be learned is formulated as production rules and presented to trainees as specific procedures during the training process. Reif (1987) found that this technique, when applied to basic science concepts, resulted in a 50% increase in learning.

A training methodology, developed by the Institute for Simulation and Training (IST) under contract to the Naval Training Systems Center, based on this cognitive science research demonstrates the training improvements that can be achieved. Application of the methodology involves using a hierarchy of goals, where each higher level goal builds upon the one preceding it. The initial process involves structuring, or engineering, curriculum lessons in accordance with the results of an explicit, detailed cognitive task analysis (Kieras, 1988). Once the instructional material is cognitively structured, additional processes such as error diagnosis and adaptive lesson frame sequencing can be implemented.

Recent research employing Naval personnel has clearly demonstrated the power of this cognitive learning approach in an applied training environment. This methodology has been shown to enhance the instructional capabilities, and therefore the effectiveness, of Navy tactical console embedded training lessons. When this technique was applied to tactical console operation, a significant improvement in learning resulted (Williams, Reynolds, Carolan, Anglin, & Shrestha, 1989, Williams, Reynolds, & Carolan, 1989; and Carolan, Williams, & Moskal, 1992). The primary goal of the present research effort was to determine whether these earlier training benefits could be replicated within the Aegis weapon system Computer-Aided Submode Training (CAST) lessons.

To help operators learn the Aegis weapons system, each console was designed with full embedded training capability. The CAST system was designed as a means of creating and presenting training lessons for each operational submode of the Aegis console. Lessons that have been created are stored on magnetic tapes. When a training session is conducted, the chosen lesson is loaded from tape and presented at the console. CAST lessons are created separately using a CAST authoring program called Lesson Generation (LGEN). Each lesson focuses on specific training objectives for a particular Aegis submode. Within a lesson, trainees can view objectives in any order, and they can backup to review previously presented material. After completing the lesson, students must complete a multiple-choice test on the factual knowledge and an advanced lesson testing the practical application of the knowledge. The advanced lesson presents a series of scenarios to the

trainee, who must perform appropriate procedures to complete the specified task. Depending on lesson performance, trainees advance to new lessons or they can review.

The Electronic Warfare Supervisor (EWS) Submode lesson was selected for use in the evaluation because it was complex enough to thoroughly test the methodology, and it was a lesson which the trainees involved in the experiment would not encounter in the course of their normal training, reducing the likelihood that they would have learned the lesson beforehand. With the original lesson selected, a systematic cognitive task analysis (Kieras, 1987a and 1987b; Williams, Reynolds, and Carolan, 1990) was performed on this lesson, which resulted in a knowledge base consisting of a network of production rules. This knowledge base served as the model of the material to be learned and as the basis for creating the instructional exercises.

The cognitive analysis methodology shares many of the principles and processes of Kieras' (1988) cognitive complexity analysis which is based upon the Goals, Operators, Methods, and Selection Rules, or GOMS model, of Card, Moran and Newell (1983). To perform a cognitive task analysis on the EWS lesson, first a hierarchy of task goals was created. Then the following tasks were completed in the order listed: specific job goals were identified within the lesson, the types of tasks that can be performed on the device which directly relate to the job goals were determined, and the particular tasks to be performed to attain those goals were specified.

Once the job goals and task goals were identified, the methods which made up the task procedures were detailed. Each method specified a goal or subgoal to be accomplished, and the sequence of steps to be executed and conditions to be met, in order to accomplish the associated task goal. Each step of a method consists of an operator or action which is executed. Operators can be perceptual, such as observing the location of a symbol; actual, such as pressing a button; or cognitive, such as making a decision, or storing or retrieving information from memory.

The result of this analysis was the detailed specification of all of the knowledge needed to complete tasks associated with console operations required of the EWS. Figure 1 illustrates a subset of the knowledge specification which was produced by the cognitive structuring process. Based on this hierarchical production system model, instructional frames were developed. Each individual rule or declarative fact in the lesson, as well as each rule that combines relevant facts and/or lower level rules, was composed into an individual exercise consisting of a frame or set of frames. In working through the frames, the student learns all the declarative facts and how these facts are interrelated. Each exercise frame created with this methodology is explicitly linked to the conditions, actions, or other rules that make up a production. This methodology is consistent with the instructional systems design principles discussed in the CAST style guide, but provides for a more specific design process. This procedure is explicitly detailed in Williams, Reynolds, Carolan, Anglin, & Shrestha, 1989.

Four primary dependent variables (DVs) were employed during this investigation, in an effort to obtain measures of learning related to performance, recognition, and recall. They were: 1) performance test errors (automatically computed at the console), 2) performance test completion time (automatically computed at the console), 3) posttest declarative knowledge score (percent correct out of 100 on the recognition questions of the posttest), 4) posttest procedural knowledge score (percent correct out of 100 on the posttest recall questions). The hypotheses for this research were as follows:

1) Subjects receiving the cognitively structured lesson will perform significantly better than subjects receiving the original lesson (DV 1).

2) Subjects receiving the cognitively structured lesson will require significantly less time to complete the performance test than subjects receiving the original lesson (DV 2).

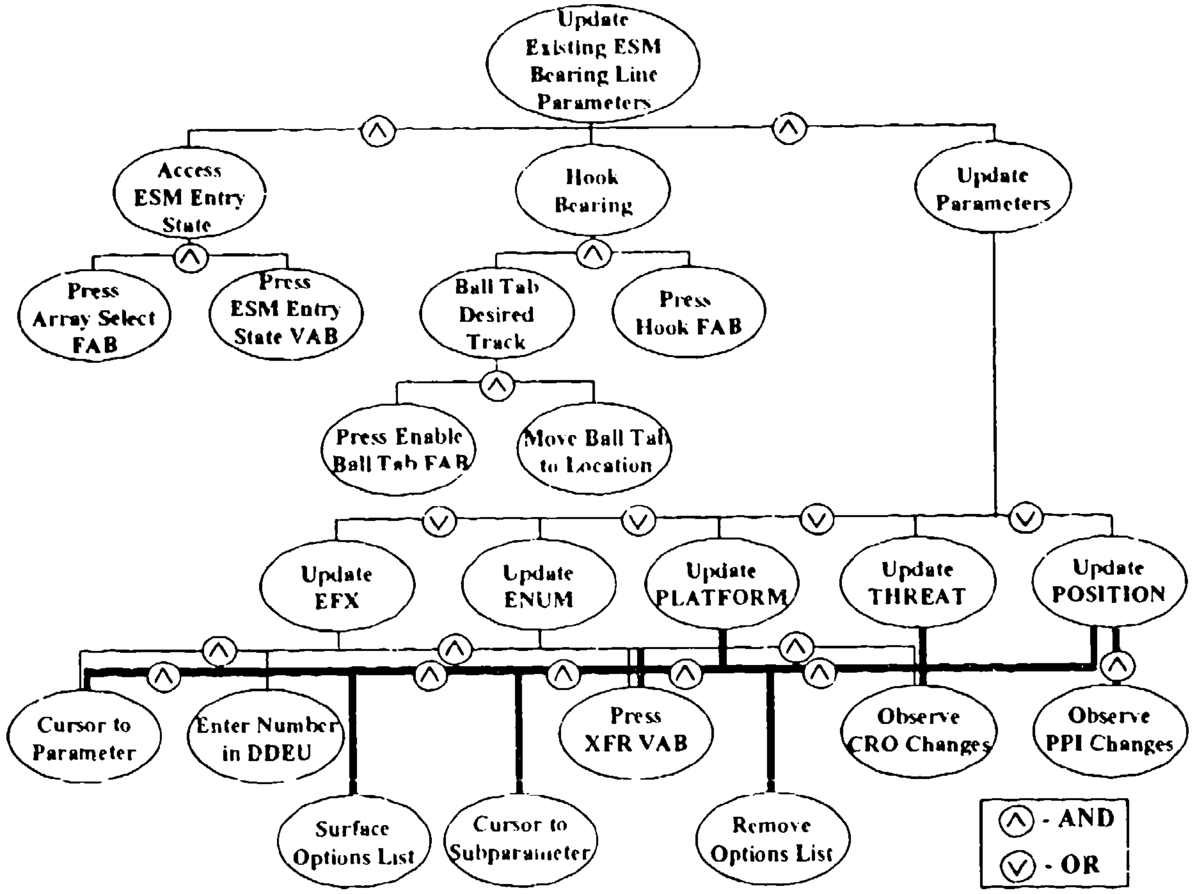


Figure 1. Task-goal hierarchy for the lesson procedure to update an EWS bearing.

3) Subjects receiving the cognitively structured lesson will remember more procedural knowledge, but not more declarative knowledge, than subjects in the original lesson condition (DVs 3-4). However, because Aegis CAST lessons are well designed according to instructional systems design principles, the benefit was expected to be less strong than with more typical Navy instruction.

EXPERIMENTAL DESIGN

This research consisted of a standard posttest-only control group design with two factors. The first condition served as the control group, in which subjects received the original lesson, which was a subset of an actual EWS Submode lesson. Subjects in the second condition received the cognitively structured lesson, based on the hierarchical production system model developed during the cognitive task analysis. The performance test was created by extracting those scenarios in the advanced EWS lesson which directly correlated with the selected objectives taught in the modified original lesson.

Subjects

Twenty-four fire controlmen participated as trainees in this research. They ranged in age from 19 to 30 with a mean of 22.6 years. Their enlisted ratings ranged from E-4 to E-7 with the vast majority being E-4. None of the trainees had prior experience with the EWS Submode CAST lesson, and few had any prior experience operating Aegis CAST lessons or the actual console in one of its operational submodes. Two trainees were eliminated because of equipment failures.

Equipment

The lessons were developed using the Lesson Generation (LGEN) program operating under the VAX/VMS operating system, using the MEGATEK WAND graphics support software. The lessons were presented on Computer Science Corporation (CSC) 451-V9 console emulators, which were controlled by the CAST Lesson Control Program (LCP) residing on a single-bay UYK-7 mainframe computer.

Procedure

The experimental evaluation was conducted in one of the Aegis Training Center's (ATC) Command Information Center mock-up laboratories. The trainee scheduling was done by Navy ATC personnel so that the class schedules of the fire control students were minimally impacted. Prior to the experiment, subjects were asked to provide background information regarding their rank and rating, age, their prior experience on Aegis consoles, CAST lessons, and specifically, the EWS CAST lesson. Subjects were then given general information about the lesson and consoles. They were told that they would spend 45 minutes working on the lesson, and that if they finished before the allotted time, the lesson would automatically restart at the beginning. Subjects were told to continue working on the lesson until the experimenter told them to stop, and they were encouraged to do their best.

Trainees were given an equal amount of time to complete the assigned lesson because pilot research showed that the cognitively structured lesson required more time to complete than the original lesson, and the researchers felt that controlling practice time was crucial. Because each subject was assigned for only one hour, 45 minutes was determined to be the maximum time that could be spent on the lesson and still complete the posttest.

Trainees were randomly assigned to either the original or cognitively structured lesson condition. Within each lesson, subjects had the capability to review any objective and to backpage to review previously displayed frames. Upon completion of the 45 minute training session, subjects were given the advanced lesson which served as the performance test, followed by the written posttest. Each trainee's performance score (number correct out of 20 items) and time to complete the test were automatically recorded by the computer

RESULTS

The means, standard deviations, and ranges for each of the four dependent measures across experimental conditions are displayed in Table 1. A qualitative inspection indicates that the data support the hypotheses, as the means obtained in the cognitively structured lesson condition are superior to those obtained in the original lesson condition. The average improvement of the cognitively structured lesson compared to the original lesson, over the four dependent measures is 47 percent.

The goal of the first two analyses was to assess the effect that lesson structure had on the ability to learn and perform procedural tasks associated with the EWS rating. Separate analyses were conducted on the performance test procedures completed and performance test completion time dependent measures. A one-tailed, independent groups t-test was conducted to assess both measures. Both t-tests proved significant ($t_{(20)} = -3.03$, $p < .01$ and $t_{(20)} = 1.90$, $p < .05$, for the performance test and completion time measures, respectively). An assessment of the strength of the treatment effects, or omega squared (w^2) was computed for both dependent measures. The omega squared was .27 for the performance test and .11 for the completion time. Cohen (1977) believes that a large treatment effect in social science research is one with $w^2 > .15$, and a moderate effect is one between .06 and .15. Therefore, our effects appeared to be fairly large.

Thus, on the performance related dependent measures, while controlling for practice time, the trainees receiving the cognitively structured lesson significantly outperformed trainees in the original lesson condition, as predicted. Therefore, the first two hypotheses were confirmed; that is, trainee performance in the cognitively structured lesson condition was significantly better than performance obtained in the original lesson condition.

Table 1. Means, standard deviations, and ranges for each DV across treatment conditions for Experiment Two (N=11 per condition).

Dependent Measures	Condition: Original Lesson	Condition: Cognitively Structured
Performance test procedures completed (out of 20)	Mean: 10.6 S.D.: 2.8 Range: 7-14	Mean: 13.6 S.D.: 2.5 Range:10-17
Performance test completion time (minutes)	Mean: 7.0 S.D.: 2.0 Range: 3-10	Mean: 5.5 S.D.: 1.5 Range: 4-8
Posttest declarative knowledge (percent correct)	Mean: 53.5 S.D.: 17.0 Range:31-88	Mean: 63.6 S.D.: 19.2 Range.25-94
Posttest procedural knowledge (percent correct)	Mean: 22.0 S.D.: 14.8 Range: 0-45	Mean: 47.4 S.D.: 16.6 Range:11-67

The multiple-choice questions on the posttest are assumed to be a measure of the trainee's ability to recognize the correct, factual information that was presented during the EWS lesson. The mean posttest score was 59% correct overall (9.4 correct out of 16). Subjects who received the cognitively structured lesson remembered more declarative knowledge than subjects in the original condition (see Table 1). However, the t-test performed on this DV was not significant ($t_{(20)} = -1.34$, $p < .10$), which corroborates previous research and our hypothesis.

The recall portion of the posttest was intended to assess each subject's ability to recall the specific procedures required to achieve a particular task goal within the EWS lesson. The average number of procedures correctly generated on the posttest, over all subjects, was 35%. Trainees in the original lesson group recalled 22% of the procedures while trainees in the cognitively structured group recalled 47.4% (see Table 1). The t-test performed on this DV was highly significant ($t_{(20)} = -3.78$, $p < .001$). The omega squared was equal to .38. Thus, as hypothesized, subjects who were in the cognitively structured lesson condition remembered more procedural knowledge on the posttest than did those who were in the original lesson.

DISCUSSION

The results support our contention that procedural knowledge and skills can be learned more effectively when structured according to the cognitive analysis methodology (e.g., Kieras, 1988; Williams, Reynolds, Carolan, Anglin, & Shrestha, 1989). Students training on the cognitively structured lesson performed better than students who received the original lesson.

Examining these data in terms of the number of subjects who reached the 80% correct criterion used by the Aegis Training Center for passing CAST lessons reveals that more of the subjects receiving the cognitively structured lesson reached criterion on the performance test and declarative knowledge section of the posttest than those receiving the original lesson (36% to 0% on the performance test and 18% to 9% on declarative knowledge). On the procedural aspect of the posttest, no one reached criterion, although eight subjects (73%) in the cognitively structured condition remembered over 50 percent of the procedures, in contrast to no one who received the original lesson.

As with our previous research (e.g., Carolan, Williams, & Moskal, 1992), the present results can be interpreted within the context of cognitive science research that makes a distinction between procedural and declarative knowledge representations. It is believed that knowledge is first encoded as facts, but the learner may not use this knowledge to carry out

procedures. Thus, to learn procedural knowledge for a particular task or setting requires that the knowledge be presented in its proper procedural context based on appropriate individual rules and the relationships among them.

Carolan, Williams, and Moskal (1992) describe in detail how to cognitively structure training materials. Suffice it to say here that this methodology provides precise direction to design and present material that should make procedural learning highly efficient. This methodology is to be used in addition to the standard instructional systems design process. Cognitive science research has shown that presenting instruction in which the specific factual knowledge, procedural rules, and the relationships among them are explicitly observable will significantly enhance learning. Structuring these lesson components in this way provides the means by which they are integrated to promote learning. Enabling trainees to combine factual knowledge with knowledge about performing associated procedures is accomplished through structured practice elements and proper feedback, which ensures that trainees know the procedures that are required and that they have properly interpreted the instructions.

Our findings confirm results obtained on similar research that we conducted for the Navy's Lesson-Translation (L-TRAN) lessons using Navy Tactical Data System (NTDS) consoles (Carolan, Williams, & Moskal, 1992). The present results demonstrate that the cognitive structuring methodology is effective in enhancing learning, even for embedded training lessons like the Aegis CAST system, which should already be efficient because they closely follow instructional systems design principles.

By structuring the lessons based on knowledge and rules that are related in an optimal manner, the first step in creating an intelligent tutoring system (ITS) that can adaptively sequence instruction has been accomplished. Examining the potential training effectiveness of an ITS in the CAST environment is recommended as the next logical step to further evaluating this cognitive structuring approach within the Navy's embedded training research and development program. This approach could be especially beneficial in this time of reduced budgets.

The cognitive structuring approach has resulted in significant learning enhancements in a variety of environments, and with trainees ranging from novices to instructors. Thus, we believe that this methodology is effective and it can be valuable for developing effective embedded training instruction. However, it does not come without costs. Foremost is the fact that implementing this methodology to restructure existing lessons is very labor intensive and time consuming. The lesson developers must become extremely well acquainted with the lesson content in order to make the proper modifications, and this takes significant time. Second, cognitively structured lessons are often longer than standard lessons, which means that more text and practice will be required. This increase may affect the available memory of the embedded training computer system, and trainees will be required to spend more time on a lesson (though they should require less remediation because more trainees will reach criterion performance quicker). However, if the lessons are created as part of an intelligent tutoring system with exercise sequencing adapted to student strengths and weaknesses, then the average lesson time could be reduced. Using an ITS should allow better students to advance through the lesson more quickly, because they would receive a shorter path with less remediation required.

The user of this approach (the organization conducting the training) must determine if the anticipated training gain is worth the additional costs and constraints. However, when initially designing procedural-based embedded training, the authors recommend developing lessons using this cognitive structuring methodology in conjunction with the standard instructional systems design approach. The costs involved with implementing the cognitive approach initially should not be significantly higher than through the traditional method, because lesson developers must be highly knowledgeable on the instructional content with all design approaches. In fact, this methodology may actually help designers create lessons more efficiently because it requires the information to be highly structured. Moreover, written user

documentation can often be created directly from the structuring process (e.g., Kieras, 1988).

REFERENCES

Anderson, J. R. (1983). The architecture of cognition. Cambridge, MA: Harvard University Press.

Anderson, J. R. (1986). Knowledge compilation: The general learning mechanism. In R. S. Michalski, J. G. Carbonell, & T. M., Mitchell, (Eds.), Machine learning: An artificial intelligence approach (Vol. 2, pp. 289-310). Los Altos, CA: Morgan Kaufmann.

Bovair, S., & Kieras, D. (1990). Toward a model of acquiring procedures from text. In Bass, Karnil, Rosenthal, and Pearson (Eds.), Handbook of reading research (Vol. II). New York: Longman.

Card, S. K., Moran, T. P., & Newell, A. (1983). The psychology of human-computer interaction. Hillsdale, NJ: Lawrence Erlbaum Associates.

Carolan, T. F., Williams, K. E., & Moskal, P. D. (1992). Structuring Curriculum for Embedded Training: An Evaluation in an Operational Setting. (NTSC Tech. Rep. 92-003), Orlando, FL: Naval Training Systems Center.

Cheng, P. W., Holyoak, K. J., Nisbett, R. E., & Oliver, L. M. (1986). Pragmatic versus syntactic approaches to training deductive reasoning. Cognitive Psychology, 18.

Chi, M. T. H., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self explanations: How students study and use examples in learning to solve problems. Cognitive Science, 13 (2), 145-182.

Chi, M. T. H., Feltovich, P. J. & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. Cognitive Science, 5, 121-152.

Cohen, J. Statistical power analysis for the behavioral sciences. New York: Academic Press, 1977.

Fong, G. T., Krantz, D. H., & Nisbett, R. E. (1986) The effects of statistical training on thinking about everyday problems. Cognitive Psyhcology, 18.

Holland, J. H., Holyoak, K. J., Nisbett, R. E., & Thagard, P. R. (1986). Induction: Processes of inference, learning, and discovery. Cambridge, MA: MIT Press.

Jeffries, A., Turner, A. A., Polson, P. G., & Atwood, M. E. (1981). The processes involved in designing software. In J. R. Anderson (Ed), Cognitive Skills and Their Acquisition. Hillsdale, New Jersey: Lawrence Erlbaum.

Kieras, D. E. (1987a). The role of cognitive simulation models in the development of advanced training and testing systems. Office of Naval Research Technical Report No. 23 (TR-87/ONR-23).

Kieras, D. E. (1987b). What model should be taught: Choosing instructional content for complex engineered systems. Office of Naval Research Technical Report No. 24 (TR-87/ONR-24).

Kieras, D. E. (1988). Towards a practical GOMS model methodology for user interface design. In M. Helander (Ed.), Handbook of human-computer interaction (pp. 135-157). North Holland: Elsevier Science Publishers.

Klahr, D., Langley, P., & Neches, R. (1987). Production System Models of Learning and Development. Cambridge, MA: MIT Press.

La Berge, D. (1973). Attention and the measurement of perceptual learning. Memory and Cognition, 1. 268-276.

Newell, A., & Simon, H. A. (1972). Human Problem Solving. Englewood Cliffs: Prentice-Hall.

Nisbett, R. E., & Kunda, Z. (1985). Perception of social distributions. Journal of Personality and Social Psychology, 48, 297-311.

Posner, M. I., & Snyder, C. R. (1975). Attention and cognitive control. In Solso, R. (Ed.), Information processing and cognition. Hillsdale, New Jersey: Erlbaum Associates.

Reif, F. (1987). Interpretation of scientific or mathematical concepts: Cognitive issues and instructional implications. Cognitive Science, 11, 395-416.

Williams, K. E., Reynolds, R. E., & Carolan, T. F. (1989). Knowledge engineering domain knowledge for an intelligent computer aided instruction system. Proceedings of the Second Florida Artificial Intelligence Research Symposium.

Williams, K. E., Reynolds, R. E., & Carolan, T. F. (1990) Automated adaptive instruction for embedded training. Proceedings of the Interservice/Industry Training Systems Conference (I/ITSC) 12th Annual Meeting.

Williams, K. E., Reynolds, R. E., Carolan, T. F., Anglin, P. D., & Shrestha, L. B. (1989). An evaluation of a methodology for cognitively structuring and adaptively sequencing exercise content for embedded training. (NTSC Tech. Rep. 89-035), Orlando, FL: Naval Training Systems Center.

COGNITIVE TASK ANALYSIS FOR DEVELOPMENT OF AN INTELLIGENT TUTORING SYSTEM

Sallie E. Gordon
University of Idaho
Moscow, Idaho

Bettina A. Babbitt
Northrop Corporation, B-2 Division
Pico Rivera, California

Herbert H. Bell
Armstrong Laboratory
Williams Air Force Base
Mesa, Arizona

H. Barbara Sorensen
Brooks Air Force Base
San Antonio, Texas

Peter M. Crane
Armstrong Laboratory
Williams Air Force Base
Mesa, Arizona

ABSTRACT

Training programs are increasingly relying on high level Artificial Intelligence modules to provide computerized feedback to trainees. The work reported here consisted of the use of cognitive task analysis methods developed at the University of Idaho to perform knowledge acquisition for a proof of concept training module targeted toward the defensive counter air mission. The specific subtask analyzed was "the use of fire control radar for search and sort" at the beginning of an Air-to-Air intercept performed by F-15 and F-16 pilots. The cognitive task methodology was conceptual graph analysis, a method that uses conceptual graphs to structure interviews and observational data gathering. The analysis consisted of three steps: (1) Development of conceptual graphs from existing documentation; (2) Expansion of the graphs through interviews structured with question probes; and (3) Expansion and completion of the graphs through performance observation and inductive analysis. After the conceptual graph analysis was completed, additional decision heuristics were used to identify the type of expert system architecture(s) most suitable for the task. These architectures include a rule-based system with explanation capability, classifiers with some type of explanation capability, and case-based reasoning with analytical ability

ABOUT THE AUTHORS

Sallie Gordon is an Associate Professor of Psychology at the University of Idaho. Dr. Gordon holds a Ph.D. in Psychology from the University of Illinois, Champaign-Urbana, and conducts applied research in knowledge acquisition and instructional design. She has recently completed a book on the ergonomic design of training programs, to be published by Prentice Hall. Bettina Babbitt received her Ph.D. in Educational Administration from Claremont Graduate School in 1982, and is the Principal Investigator for Independent Research and Development projects in advanced training systems for Northrop Corporation. Herbert Bell is a Research Psychologist with the Air Force Armstrong Laboratory at Williams Air Force Base. He received his Ph.D. in Experimental Psychology from Vanderbilt University, and his current research concerns the use of distributed interactive simulation for combat mission training. H. Barbara Sorensen holds a Ph.D. in Instructional Design and Computer Technology, and leads the Joint Service Technology Initiatives in the Plans and Programs Division of the Human Resources Directorate, Air Force Armstrong Laboratory. Peter Crane is a Research Psychologist with the Air Force Armstrong Laboratory. He received a Ph.D. in Experimental Psychology from Miami University (Ohio), and his current research is focused on identifying the training opportunities and instructional strategies for networked, multiple aircraft simulations.

COGNITIVE TASK ANALYSIS FOR DEVELOPMENT OF AN INTELLIGENT TUTORING SYSTEM

Sallie E. Gordon
University of Idaho
Moscow, Idaho

Bettina A. Babbitt
Northrop Corporation, B-2 Division
Pico Rivera, California

Herbert H. Bell
Armstrong Laboratory
Williams Air Force Base
Mesa, Arizona

H. Barbara Sorensen
Brooks Air Force Base
San Antonio, Texas

Peter M. Crane
Armstrong Laboratory
Williams Air Force Base
Mesa, Arizona

INTRODUCTION

Expert systems are increasingly being used in computer-based training programs as a way to efficiently capture the expertise of instructors and other personnel, and provide that expertise to students. In most cases, a computer-based tutorial is first developed and then a standard rule-based expert system is embedded as a module within the system to provide task-related information and performance feedback at appropriate points.

While expert system technology can provide useful tutorial mechanisms, it has been noted that traditional rule-based expert systems have certain drawbacks. One of the most critical is the fact that they often have inadequate explanation capabilities (e.g., Gordon, 1992). Because of this and other limitations (e.g., "brittleness" and the difficulty of obtaining rules from experts), new types of expert systems are under development that have a broader range of capability. These includes systems such as neural networks, fuzzy logic systems, and deep model-based systems (Gordon, 1991). These new systems may be more appropriate for certain types of training programs than more traditional rule-based systems.

In summary, training systems that must capture some element of cognitive expertise can rely on expert systems as a mechanism, but there are certain issues involved in their implementation. One is the question of which expert system technology is appropriate for a given project. Gordon (1991) recently published a heuristic for determining which of the expert system technologies would be most appropriate for a given task, depending on certain characteristics of the task and user. While this heuristic was developed for the use of expert systems as a general class of tools, we hypothesized that it might be equally applicable for determining the appropriate expert system technologies for a given training application.

Among other types of information, the decision heuristic for identifying the appropriate type(s) of expert system requires identification of the types of knowledge primarily used in the task. This means that a cognitive task analysis must be performed before implementing the heuristic. Since some type of task analysis should be conducted to acquire the knowledge base for a training program anyway, this step does not constitute an additional requirement.

PROGRAM OBJECTIVES

The project described in this paper is a multiyear proof of concept endeavor. For the first year, there were three objectives. The first was to identify a task that is primarily

cognitive for which part-task training could be implemented. The second was to perform an in-depth cognitive task analysis using the conceptual graph analysis methodology recently developed at the University of Idaho (Gordon & Gill, 1992; Gordon, Schmierer, & Gill, in press). The third was to use the results of the cognitive task analysis as input to the expert system decision heuristic. This would allow us to evaluate the usefulness of the heuristic for choosing expert system technologies within the specific context of a computer-based training program.

The three tasks corresponding to these objectives will be described below. However, before describing the work performed for the project, we will provide a brief overview of the basic cognitive task analysis method, conceptual graph analysis (CGA).

CONCEPTUAL GRAPH ANALYSIS

The CGA method consists of using several knowledge acquisition techniques to develop a knowledge base in the form of one or more conceptual graph structures (Gordon et al., in press). The knowledge acquisition techniques vary, but usually consist of the following steps, done in the order listed, although one may iterate through steps 2 and 3 several times:

1. Document analysis
2. Structured interviews; question probes
3. Observation and inductive analysis.

Before describing each of these methods, we will briefly review the knowledge representation syntax upon which the method rests.

Knowledge Representative Via Conceptual Graph Structures

Conceptual graph structures are a type of concept graph or network based on a highly specific graph syntax. They are most easily described as a combination of semantic networks, propositional networks, and goal hierarchies (e.g., see Gordon & Gill, 1992; Graesser & Gordon, 1991).

Conceptual graphs consist of nodes linked by labeled, directional arcs. Figure 1 shows an example of a small, incomplete graph for

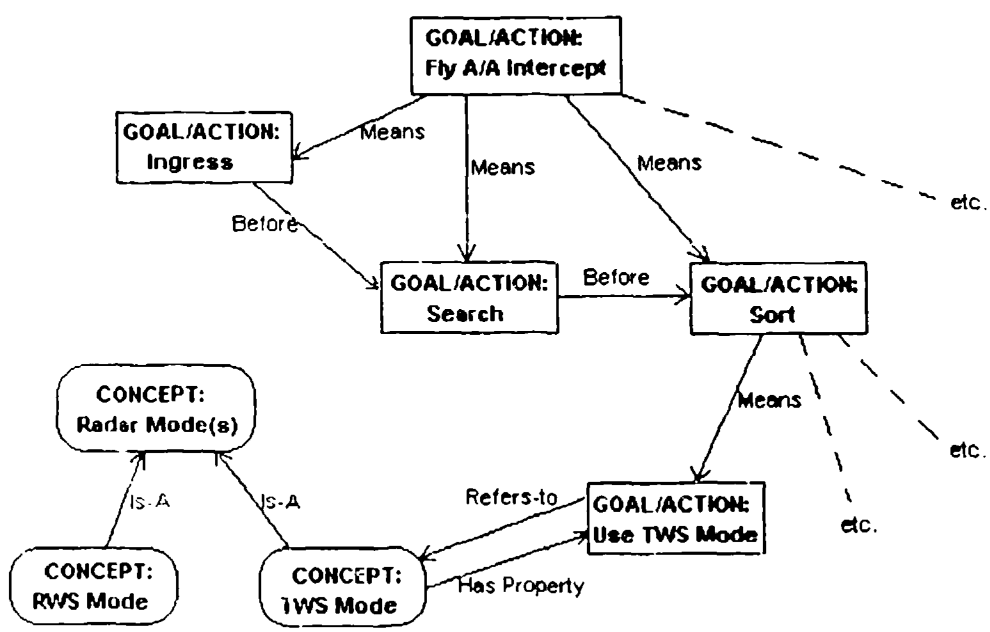


Figure 1. Small and incomplete conceptual graph structure with concepts and goal/actions relevant to sorting task.

information relevant to the F-16 radar sorting task used in this study.

Each node in a graph contains two types of information; the specific content of the node (e.g., TWS Mode) and the category or type of information contained in the node (e.g., Concept, State, Event, Goal/Action, etc.). The categorization of the information in nodes preserves information about the types of knowledge and relationships, helps organize the information into substructures, and provides support for the question probe method (described shortly).

Unlike other graph methods such as concept mapping, conceptual graph structures are based on a specific and well-defined set of arcs that relate the nodes. The most frequently used arcs are listed in Table 1; organized by the type of substructure in which one typically finds them. That is, a body of knowledge may consist of many types of knowledge, such as functional system components, goal hierarchies containing information about how to use the system, etc. These different types of knowledge tend to cluster into subgraphs, but the subgraphs are also interrelated with one another, as shown in Figure 1.[1]

It can be seen that conceptual graph structures can be used to represent a variety of types of knowledge, including semantic knowledge, knowledge of structural systems such as an automobile or jet aircraft, knowledge of causal systems such as how various factors interrelate in systems such as an engine, the human body, the physical environment, etc., and knowledge of complex procedures such as using controls and displays for controlling a vehicle.

In addition to general or "semantic" types of knowledge, conceptual graphs can be used to represent more specific information. Examples might be specific instances of categories such as doctors we have known or cars we have owned. Each instance is associated with its more general term via a Has Instance arc (or conversely, by an Is-Instance-of arc). Episodes we have experienced are likewise associated with relevant nodes in the network. Similarly, visual or auditory information is assumed to be associated with parts of the network. In representing this type of information, we generally include labels for the information in

[1]Readers are referred to Gordon & Gill, 1992, or Gordon, Schmierer & Gill, in press, for a more in-depth presentation of this material. In addition, a tutorial on conceptual graph structures is available from M. DeVries and D. Sorensen at University of Idaho.

Table 1. Conceptual graph substructures and arcs commonly used within the substructures.

TAXONOMIC STRUCTURES: Specify the relationships between superordinate and subordinate concepts (e.g., Apple Is-A Fruit).

- Is-A
- Has Property
- Has Instance
- Has Part
- Refers-to
- And/Or

SPATIAL STRUCTURES: Contain knowledge delineating the spatial layout of regions and objects in regions.

- Above/Below
- Left-of/Right-of
- Behind, etc.

CAUSAL NETWORKS: Contain knowledge about causally driven state and event chains.

- Has Consequence
- Manner
- Before/During/After
- And/Or

GOAL HIERARCHIES: Specify goals, cognitive activities, and behavior procedures for accomplishing goals.

- Reason/Means
- Initiates
- Before/During/After
- Manner
- Has Consequence
- And/Or

---

the actual network itself (e.g., "image of Dr. Smith").

The specific information as well as visual and auditory nodes is important because sometimes people rely on direct associative learning to perform a task. For example, I might associate a range of images with the concept of "too dark" for toast. There is no specific conceptual or semantic rule for too dark, just a set of instances. If I make judgments in the future by comparing new instances with old instances, this "expertise" can be captured directly within the network by using instances rather than by trying to derive more general rules.

In summary, conceptual graph structures are used to represent a variety of types of knowledge. We can say that they capture verbalizable, declarative knowledge via the four types of subgraphs listed in Table 1. They also capture the more difficult to verbalize procedural knowledge by representing specific associations between stimulus sets and responses (e.g., the toast example).

Methods for Performing the Conceptual Graph Analysis

The task analysis method termed Conceptual Graph Analysis consists of a complementary set of methods for eliciting, representing, and analyzing a body of knowledge using conceptual graph structures as the representation medium. The methods have been chosen because, used together, they maximize the amount and types of knowledge that will be elicited during the task analysis. They are especially appropriate for use with experts who may be able to verbalize only part of their domain knowledge. The methods are briefly described below (see Gordon et al, in press, for a more detailed presentation).

Document Analysis. The first step in conceptual graph analysis usually consists of identifying relevant information in documents and translating the information into conceptual graph form.

In developing conceptual graph structures from documents, one typically encounters several problems. Among others, these include three critical ones. First, much of the information written in a prose format contains semantic ambiguities. For example, when one determines to graph the statement;

"hold the valve open and loosen the nut,"

it is unclear whether one performs these tasks at the same time, or one before the other Readers are unaware of the extent to which ambiguities exist in written material because they use their own knowledge to interpret the material. However, sometimes ambiguities are problematic for the reader. When conceptual graph structures are developed from documents, there are often ambiguities which the researcher cannot resolve alone. These instances are noted and addressed in the next interview phase.

A second problem that will become noticeable during the process of graphing document information is that certain information will be missing. For example, the document might instruct a person to perform some task but not include relevant information regarding how to perform the task, or more frequently, omits necessary information on the conditions under which one should perform the task (when to perform the task).

The problem just described can be thought of one where there are missing nodes in the conceptual graph structures (either single nodes or whole branches). A third related problem is that while there may be a sufficient number of the appropriate nodes, the relationships between them are lacking to some degree. This is a frequent occurrence in science and engineering textbooks. For example, the author will describe a principle, and next describes a problem along with the steps needed to solve it. But the author fails to adequately describe the relationships between the basic principles and the problem steps (i.e., goal hierarchy).

When people have trouble with instruction manuals, it is almost always because either information is ambiguous or it is missing. The expert who wrote the manual was so familiar with the task domain, that they couldn't see these deficiencies. Graphing the information makes it "visually salient" that the information is missing. For example, each goal/action node should have subordinate nodes (via Means arcs) and initiating nodes describing the circumstances under which one performs the goal/action.

Structured Interviews. Documents almost never contain all of the information needed to create complete and conceptually coherent conceptual graphs. Therefore, after the graphs have been initialized using document analysis, the next task is to expand and clarify the graphs by consulting the domain experts. This is usually done through structured interviews.

Interviews with one or more SMEs are structured through the use of question probes (Gordon & Gill, 1992). Question probes are questions that elicit knowledge relevant to each node on the conceptual graph structure. Question probes are specific questions that the researcher asks for each node on the graph. Each node type has its own unique set of questions. For example, a Concept node would result in the researcher using questions such as:

What is ____________?

What are the properties or characteristics of ____________?

What are some instances of __________?

Thus, the Concept node of "TWS mode" would result in questions such as:

What is TWS mode?

What are the properties of TWS mode (that is, what happens in TWS mode)?

etc.

For most of our applications, the researcher takes the conceptual graph structure to the interview session (the graphs are actually divided into several subgraphs to make them more manageable). These graphs are used as a visual job aid for the researcher and expert to examine. The question probes are given to the expert, and answers are tape-recorded. The interviews usually go into great detail about all types of information; conceptual knowledge, tasks and how they are performed, the conditions under which one performs subtasks, and so forth. This process usually takes numerous interviews, and the expert frequently looks at the graphs to remind him or herself what has been said previously.

Observation and Inductive Analysis. Most of the information can usually be obtained through interviews structured with question probes. However, experts often perform tasks without really knowing how or why. When they have trouble verbalizing this information, it is then necessary to have them perform the task under a variety of circumstances and record the stimulus conditions and resultant actions.

Therefore, this step consists of asking SMEs to perform the primary task under a wide variety of circumstances. Think aloud verbalization is not required, although they are encouraged to do so if it does not interfere with their performance. Audio tapes or videotapes are made of task performance and the researcher reviews these tapes afterwards. In many cases, the expert and researcher review the tapes together. Through review, rules associating situational cues with specific decisions and behaviors are induced. These rules are validated against other instances of task performance. Occasionally, it is not possible to identify a specific set of rules to account for the expert behavior. In this case, the situational cues and associated actions are represented directly as instances in the graphs.

Rational Analysis. While the previous three methods are the means by which we acquire the information to go into the conceptual graph structures, ideally the analyst will also spend some effort evaluating the conceptual graph structures for clarity, completeness, logical consistency, etc. This has several functions. First, it can place less of a burden on the expert to perform this function. Second, while the expert may have described his or her particular method for accomplishing a goal, a rational analysis of the system components and functional relationships might yield a more efficient or effective method.

Advantages of Conceptual Graph Analysis

Conceptual graph analysis has several advantages over other methodologies. In addition to receiving empirical support (Gordon et al., in press), it has now been used for knowledge acquisition and task analysis in over a dozen different domains (e.g., forest management, using a literature search system,

using a VCR, engineering mechanics and problem solving, teaching cooking skills to cognitively disadvantages learners, etc.).

It can be seen that one advantage is that it is domain-general. Other advantages include:

1. Unlike other syntaxes such as GOMS or concept maps, it can be used to represent and integrate all major types of knowledge including general taxonomic knowledge and goal structures.

2. The graphs provide a standardization of representation, a useful shorthand for interviews, and a visual means to see interrelationships among concepts.

3. The graphs yield and support question probes for structuring interviews.

4. Question probes are a simple to use but powerful method for pressing the questioning process into incomplete areas of the knowledge base.

5. The method integrates several complementary means for knowledge acquisition, which results in acquisition of both verbalizable and implicit or procedural knowledge.

METHOD AND RESULTS OF TRAINING ANALYSIS RESEARCH EFFORT

For the project described in this paper, the work was performed in three steps corresponding to the three objectives noted earlier. For each step, we will briefly describe the method and results obtained.

Identify Cognitive Subtask

There were several constraints that needed to be met during the process of identifying the cognitive subtask to be used in this study. These were:

- Since the work was largely to be performed at Armstrong Laboratory, Williams Air Force Base, the task had to be one that could be studied at that site.

- The Air In[illegible]pt Trainer (AIT) was available for [illegible]ervational data collection ([illegible]equired by conceptual graph analysis). This system is a relatively high level computer-based simulator for pilots to practice air intercepts against 1-5 targets. Therefore, the task had to be one that pilots could perform on the AIT.

- The task had to be one such that expert pilots were available either from Williams or Luke AFB to act as subject matter experts (SMEs).

- The subtask itself and knowledge used to perform the task must be unclassified.

- The task must be primarily cognitive.

- The task must be relatively limited in scope and unrelated to other tasks.

The cognitive subtask chosen for the training program consists of using the F-16 radar to "develop the big picture" during an air intercept. In other words, the process of searching a given airspace and developing a mental model of aircraft activity within that space. Subtasks include searching for targets using the F-16 radar search mode, evaluating the nature of target aircraft activity using additional radar modes, and developing a complete and accurate mental representation of the air space before deciding on a group to target. For this particular project, the task was further constrained such that a pilot is performing these subtasks without help or outside communication.

In standard terminology, this task essentially consists of using the radar for searching and sorting. While not quite accurate, for the sake of expediency, in this paper we will refer to the task as "sorting."

Cognitive Task Analysis

Conceptual graph analysis was used to carry out the cognitive task analysis.

Document Analysis. The first step in the analysis consisted of translating existing documentation relative to the subtask into conceptual graph structures. Working at Armstrong Laboratories, a dozen documents were reviewed and all information relevant to the task and its subtasks was translated into conceptual graph structures. The documents included basic conceptual booklets on the fire

control radar system and training manuals for performing air intercepts. Approximately 12 conceptual subgraphs were first drawn on drafting velum (to be able to see the big picture), and then converted to one large computer-based network using a special version of "SemNet" (Fisher, Saletti, Patterson, Thornton, Lipson, & Spring, 1990) on a Macintosh personal computer. This program displays all of the nodes and arcs in either graphic or list format. It program also has several search and traversal mechanisms.

The document analysis took approximately 80 man-hours; much of this time consisted of identifying the specific subtasks to be included and carefully combing through the documents to find applicable material. Most of the information obtained from the documentation pertained to the radar system: it's functional components, descriptions of radar modes, etc. While a moderate degree of information was also found for flying intercepts, little was found for how to use the various radar modes for searching and sorting.

Structured Interviews. Interviews were carried out with nine SMEs consisting of F-16 pilots, F-15 pilots, and instructor pilots. For each SME, the (paper-based) graphs were evaluated to determine what information was inconsistent or missing. The SMEs were given question probes based on the graphs (see Gordon & Gill, 1992) to obtain the necessary information.

In the interviews, pilots described their use of the various radar modes for searching and sorting activities. Some pilots focused mostly on strategies for carrying out the intercept, and did not seem to focus substantially on the radar modes. Other pilots discussed radar modes in conjunction with other strategy information, and indicated that various modes are most appropriate only under certain circumstances. All interviews were tape-recorded, and the information was subsequently added to the graphs. One pilot (F-15) had goal hierarchy information that was substantially different from the other SMEs; this information was translated into a separate graph.

The graphs were greatly expanded through the structured interview process. The interviewing and graphing processes took somewhere between 120 and 160 man-hours on the part of the researcher. The combined graphs at this point had approximately 1100 links. While much information was gained, it was apparent that some of the task was performed using perceptual or "implicit procedural" knowledge not easily verbalized. For this reason, observation with inductive analysis was next performed.

Observation and Inductive Analysis. For this part of the task analysis, two F-16 pilots and one F-15 pilot performed numerous air intercepts on the AIT simulator. In each scenario, they were required to search for targets and determine the number of groups, number of individual targets, and location of all targets. They then decided on one group/plane to intercept, and flew the simulator in for the intercept. Once it became apparent that they either would or would not make the intercept, the scenario was terminated. All scenarios were videotaped and the radar screen was also recorded separately for clarity in the review process. In this way, we were able to determine the general activity of the pilot as well as his specific use of the radar at all times.

It was apparent from the performance of the three pilots that the F-15 pilot was the most expert at using the radar and its various modes for searching and sorting. For this reason, the F-15 pilot was asked to participate in follow-up reviews of the tapes and additional structured interviews. During reviews of the tape, the expert gave explanations for cognitive and behavioral activity during the scenarios. These review sessions were tape-recorded.

Based on researchers' analysis of the videotapes as well as the expert reviews of those videotapes, additional information was added to the conceptual graph structure. Most of this was of the goal hierarchy type of information; what to do, for what reasons, and under what circumstances. In creating a traditional expert system, this information becomes the IF-THEN rules of the system.

The observation, retrospective inductive analysis, and expansion of the graphs took approximately 120 hours on the part of the researcher. At this point, the combined graph had approximately 1500 links.

Findings From The Cognitive Task Analysis.

The conceptual graph analysis resulted in a large and useful conceptual graph structure. The use of several SMEs revealed the fact that while all pilots were experts at their job, some were more expert than others at using the radar modes for searching and sorting. In particular, the F-15 pilot was extremely adept at this task, undoubtedly because it is a more central part of that job (Air-to-Air Intercepts) than the F-16 pilot who focuses more frequently on Air-to-Ground missions.

The conceptual graph structures showed that some of the knowledge used for searching and sorting is verbalizable declarative knowledge, while some of it is a more difficult to verbalize implicit/procedural type of knowledge. However, this latter type of knowledge was simple enough in structure that we were able to induce the concepts and interrelationships from behavior, and make it explicit in the graphs.

Table 2. Potential architectures for expert systems used in training programs.

SYMBOLIC/ANALYTIC
- Traditional rule sets
- Rule sets with explanation capabilities
- Fuzzy logic systems
- Static classifiers
- Classifiers with genetic algorithms
- Model-based systems

CONNECTIONIST
- Neural networks
- Neural networks with connection weights for explanation

CASE-BASED
- Case-based reasoning, retrieval
- Case-based reasoning, analytical ability

Front-End Analysis: Determining the Appropriate Expert System Technology

Once the conceptual graphs were complete and the performance data base obtained, it was possible to analyze the task with respect to the underlying cognitive characteristics. Gordon (1991) reviewed several expert system technologies and suggested that certain characteristics of the task along with user needs will determine the most appropriate expert system technology. The technologies are shown in Table 2.

The major task characteristics used in the decision heuristic include:

- Whether it is a stable or unstable stimulus environment (do stimuli and therefore decision rules undergo change over time)
- Whether there is subjective judgment (preference)
- Whether the task knowledge base is narrow or broad and complex
- Whether the task knowledge base is well-defined or ill-defined
- Whether expert performance is rule based, analytical, perceptual, or a combination

Application of the heuristic consists of identifying where the particular domain lies on several dimensions and then using the Table published in Gordon (1991) to identify the appropriate expert system technology.

The task analyzed in this project is characterized by a dynamic environment where information is obtained by the pilot in a sequential manner. The knowledge base is large and complex (many interrelationships), but relatively well-defined. Expert performance is based on analysis, rules, and perceptual performance. In addition, the time to perform the sorting task is extremely short. This last characteristic suggests that two training approaches using expert systems are possible: either feedback from the system as the pilot is performing the sorting task, or feedback after the pilot has performed the sort.

Consider the first case, feedback during task performance. In this case, certain types of system are not likely to be appropriate, for example, a case-based reasoning approach is not likely to be helpful because the time

required for the pilot to process the case and make an appropriate conversion is limited.

A rule-based expert system might be used if hardware allowed direct access to situational cues and timely provision of an answer. Other acceptable systems include classifiers with or without genetic algorithms. The major drawback with this approach is that the provision of additional explanatory information would prove disruptive.

More likely, the trainee would perform the task on a simulator, without guidance, and then an expert system would be consulted for feedback. To determine the most appropriate expert system technology, the following analysis was performed:

1. For the task of sorting, the answer cannot be gained from analysis of a complete system model, therefore the "model-based system" approach is not appropriate.

2. The task is characterized as ill-defined, complete, and relatively narrow. The task is performed by a novice-trainee, who presumably would want explanations to accompany "answers." This points to the use of one of the following types of expert system:

- Rule-based with explanation capability
- Classifiers with some type of explanation capability
- Case-based reasoning with analytical ability

Because the task is information or verbal-based as well as perceptual, a neural network alone would probably not be the best choice.

SUMMARY AND CONCLUSIONS

The cognitive task analysis method, conceptual graph analysis, was successfully applied to the defensive counter air mission subtask of "sorting" using the fire control radar system. We were able to identify the goals and actions required to perform the task, the various conditions for alternative methods of performing the task, and the situational cues used to choose among the alternative subgoal hierarchies. Pilots were able to verbalize most of this information in great detail during question probe sessions. However, it was also necessary to observe actual task performance to identify some of the conditions under which pilots used various strategies. For the SME who was used most heavily for the analysis, it was helpful to intersperse performance sessions with follow-up structured interviews. We found that the SME performance was very consistent with rules verbalized via interviews.

The conceptual graph analysis resulted in a knowledge base in network format, with approximately 1500 links. This knowledge base will yield the information required to develop a cognitive part-task trainer with an embedded expert system.

Analysis of the cognitive characteristics of the task, as well as analysis of user needs, was successfully performed, and suggested several appropriate expert system technologies. This is a preliminary indication that the heuristic developed for expert system selection was adaptable to the context of instructional system design. The next step will be to implement the system as an actual training program.

REFERENCES

Fisher, K. N., Saletti, J., Patterson, H., Thornton, R., Lipson, J., & Spring, C. (1990). Computer-based concept mapping. Journal of College Science Teaching, 19, 347-352.

Gordon, S. E. (1991). Front-end analysis for expert system design. Proceedings of the Human Factors Society 35th Annual Meeting (pp. 1412-1416). Santa Monica, CA: Human Factors Society.

Gordon, S. E., & Gill, R. T. (1992). Knowledge acquisition with question probes and conceptual graph structures. In T. Lauer, E. Peacock, and A. Graesser (Eds.), Questions and information systems (pp. 29-46). Hillsdale, NJ: Lawrence Erlbaum Associates.

Gordon, S. E., Schmierer, K. A., & Gill, R. T. (in press). Conceptual graph analysis: Knowledge acquisition for instructional system design. Human Factors.

Graesser, A. C., & Gordon, S. E. (1991). Question answering and the organization of world knowledge. In G. Craik, A. Ortony, and W. Kessen (Eds.), Essays in honor of George Mandler (pp. 227-243). Hillsdale, NJ: Lawrence Erlbaum Associates.

Virtual Time:
Adding the Fourth Dimension to Virtual Reality

Dutch & Elizabeth Guckenberger
ECC International Corp./UCF
Kay Stanney, UCF, Dan Mapes, IST/UCF
Orlando, FL

The Virtual Time (VT) concept is an unique new manipulation of time in the context of Virtual Reality. VT refers to a Virtual Reality paradigm that manipulates time under the control of the operator, instructor, or software. Current Virtual Reality environments allow operators to control space. Virtual Time extends operator control to vary the flow of "simulated time", that is, "Time-Warp" the virtual environment. A hypothesis of the immersive nature of Virtual Reality which tightly binds an individual's "time norm" to the speed of environmental cues is presented and provides the framework within which to define the VT concept. The pilot study presented in this paper can also be characterized as the first use and extension of the Above Real-Time Training (ARTT) paradigm. In this application of Virtual Time, twenty-eight university students performed a simple tracking and targeting task under two levels of time compression, (i.e., 1.0x, 1.7x). All subjects were then tested in a real-time (1.0x) environment. This study investigated a virtual block grabbing task. The block moved in a three dimensional virtual environment and subjects were required to use a Virtual Reality glove to track and grab the block. In the block grab task the mean performance for the VT (1.7x) trained group performed twice as fast as the control group (1.0x) during testing (transfer of training) when both groups were tested at real time. Post test, a set of questionnaires were administered to subjects in order to establish the perceived temporal and workload demands of the task. The results from these questionnaires indicated that within both subject groups (1.0x and 1.7x), there were no significant differences detected between the perceived temporal and mental demands of the testing and training phases. This indicates that the VT group did not perceive the change in temporal demands between the training (1.7x) and the testing (1.0x) phases. There were, however, significant differences in the perceived temporal demands between subject groups. The VT group perceived less temporal demands during the testing (1.0x) phase than the control group. These results indicate that VT is a potential means of exploiting an existing ability of humans (time adaptability) within virtual training environments in order to achieve performance enhancement in real-time situations. ARTT analogies and parallel concepts are discussed including a synthesis of multi disciplinary support for Virtual Time. Conclusions and novel future research directions are presented.

Dutch Guckenberger: Dutch Guckenberger received his M.S. is Simulation Engineering and his B.S. in Computer Science & Physics from the University of Central Florida (UCF). He currently teaches graduate courses in real-time simulation and is pursuing his Ph.D. in IEMS: Simulation Systems at UCF. Mr. Guckenberger is the Senior Software Design Engineer for ECC International Corporation's R&D Department whose specialties include real-time graphics, firmware and human sensory abilities. He has been involved with simulation for 11 years and has concentrated on ARTT research for the past 4 years. He is member of IEEE, ACM, and SIGRAPH.

Dr. Kay Stanney: Dr. Stanney, an assistant professor at the University of Central Florida, is a graduate of the School of Industrial Engineering at Purdue University. She is currently developing a collaborative Virtual Reality (VR) effort at UCF, whose objective is to facilitate the wide-spread use of VR technology. She is also involved with ergonomic research at NASA KSC, where space shuttle operations are being analyzed in terms of processing and scheduling efficiency. Dr. Stanney's primary areas of research include human-computer interaction and cognitive engineering. She is a member of IIE and the Human Factors Society.

Daniel Mapes: Daniel Mapes received his B.S in Physics from Carnegie Mellon University and is currently working on his M.S. in Computer Science. Although actively involved in the development of Virtual Environments (VE) products at IST, he is also co-founder and president of a VR Research Team or "Toy Scouts™", a group of volunteer students which forms an educational link between high school, college and local industry in the development of VE skills.

Virtual Time:
Adding the Fourth Dimension to Virtual Reality

Dutch & Elizabeth Guckenberger
ECC International Corp./UCF
Kay Stanney, UCF, Dan Mapes, IST/UCF
Orlando, FL

INTRODUCTION

The Virtual Time (VT) concept is a unique new manipulation of time in the context of Virtual Reality. VT refers to a Virtual Reality (VR) paradigm that manipulates time under the control of the operator, instructor, or software. Current Virtual Reality environments allow operators to control space. Virtual Time extends operator control to vary the flow of "simulated time", that is, "Time-Warp" the virtual environment. This pilot study can be characterized as the first use and extension of the Above Real-Time Training (ARTT) paradigm, to a Virtual Reality environment (Guckenberger,et al, 1992). Prior successful research results with ARTT on simulators encouraged the extension of time manipulation to a Virtual Reality environment. The hypothesis of this study is that the immersive nature of virtual environments will bind an individual's time norm to the speed of environmental cues presented in the VR environment.

In prior research using ARTT on tank gunnery, (Guckenberger, et al, 1992), post test questions indicated that subjects were unable to distinguish time compression changes during the experiment. Subjects were unable to distinguish changing from a 1.0x to a 1.7x condition and were also unable to distinguish a 1.7x from a 2.0x condition. Additionally, post test comments from an ARTT study with US Air Force pilots on part task F-16 trainers further supported the hypothesis that subjects are unable to distinguish between less than 2X changes in time compression. Both of these studies measured the perceived temporal demands via an informal- post-task query.

An attempt was made in this study to use well established and validated scales to measure perceived temporal demands and to determine if the time norm hypothesis extends to the virtual environment.

BACKGROUND

There is a growing body of research showing the "Time Adaptability" of Man (Holubar, 1962, Kolf, 1973, Hoey, 1976, Vidulich, 1983, Matin & Boff, 1988, Guckenberger, et al, 1992). Virtual Time (VT) is a method of exploiting an existing ability of humans (time adaptability) with existing capacity in Virtual Reality environments (i.e. software only changes to virtual form and function). The VT concept can be characterized as a synthesis of emerging man-machine interface technologies that manipulate time (i.e., RAP-COM, ARTT). Virtual time investigations are based upon human time perception and can be viewed as an extension of Above Real-Time Training research.

Psychophysical research into time perception has shown the relativistic nature of time perception in humans. (Jones, 1976, Toumodge, 1990, and Skelly, 1993). Relativistic nature is defined as linking an individual's perception of time to his/her "stimulation state" or "time norm". This is analogous to Einstein's theory of special relativity which links relative velocities to a particular observer's frame of reference. It is noteworthy that this analogy was arrived at independently by individuals in three different fields (Jones, 1976, Toumodge, 1990, and Guckenberger, et al, 1992). Working models of relative perception in audio training have been proposed (Hahn & Jones, 1981). Current work is being done to extend these audio findings to the arena of visual training (Skelly, 1993, Guckenberger, et al, 1992). These studies provide VT and ARTT with a firm theoretical basis upon which to build. These studies indicate that time perception can be altered if a particularly boring or interesting task is introduced, or if the arousal state of the subject is changed through external environmental cues (Parasuraman, 1986). Humans perceive time differently depending upon the individual's "stimulation state" or "time norm". This stimulation state is based, in part, on the sensory cues in the environment and the interactivity level between the individual and his or her environment.

Virtual environments, multi-sensory worlds which can be manipulated by users, provide highly interactive experiences. It is therefore reasonable to suggest that the immersive nature of Virtual Reality, coupled with virtual time, can alter an individual's "stimulation state" or "time norm". It is also suggested that the resulting perception of time elicited by a particular stimulation state forms a "time frame of reference" for that individual. If the stimulation environment is altered, the individual's time frame of reference will correspondingly recalibrate (without the individuals conscious awareness) in order to accommodate the new time demands of the environment.

When an individual's subjective time reference is perceived as long, it may offer a unique advantage for providing training on critical high performance skills. This artificially accelerated frame of reference may give the operator more "perceived time" in which to actually perform key elements of the mission. It may be suggested that the very perception (i.e. realization) that the operator has more time may lead to better decision making and situational awareness. It may give the operator the edge that makes the difference in today's modern battlefield. Due to the fact that VT training occurs in the same exact environment as real-time testing (i.e., the task stimuli and required responses are the same; time is the only variable manipulated), no negative transfer should be expected (Holding, 1965). In fact, due to the similarity between the task stimuli and required responses, a high transfer between training and testing should be expected. This means that more economic training can occur on existing VR simulators by accelerating the internal time flow. The simplest case for VT is improved VR simulator usage either by more trials per unit time per trainee, or higher trainee throughput.

ARTT HISTORY

Time compression has been studied with ARTT in flight simulators. Above-Real-Time Training (ARTT), refers to a training paradigm that places the operator in a simulated environment that functions at faster than normal time. In the case of air combat maneuvering, a successful tactical air intercept which might normally take five minutes, would be compressed into two or three minutes. All operations of the intercept would correspondingly be accelerated such as airspeed, turn and bank velocities, weapons flyout, and performance of the adversary. In the presence of these time constraints, the pilot would be required to perform the same mission tasks to the same performance criteria--as he would in a real time environment. Such a training paradigm represents a departure from the intuitive, but not often supported, feeling that the best practice is determined by the training environment with the highest fidelity. ARTT has been implemented economically on existing simulators. It is important to realize that ARTT applications require the simulated velocity of the targets and other entities to increase, NOT the update rate. Over 25 years ago, NASA flight test engineers recognized that if one could program a simulator to operate in "fast time", one could give test pilots a more accurate experience or "feel" of real-word stresses that would be present in the aircraft (Kolf 1973, Hoey 1976).

The origin of support for ARTT, in simulators, comes from anecdotal reports from NASA. Researchers at the NASA Dryden Flight Research Center during the X-15 program in the late 1960's needed a mechanism to address the X-15 test pilots' post flight comments of being "always behind the airplane..." and "... never catching up" (Thompson, 1965). Clearly, there were some differences between the perceived time in the well-practiced simulator flights and perceived time in the experimental aircraft. The first time NASA used fast time simulation was toward the end of the X-15 program. Pilots compared practice runs at various time constants with flights they had already flown. A fast time constant of 1.5x felt closest to their flight experience and was planned on being implemented in the lifting body programs. Lack of funding precluded the program from fully developing the capability, however, NASA's test pilots at DFRC have endorsed the benefits of using "fast time" simulation as part of the training process (Kolf 1973, Hoey 1976).

Past studies (Vidulich, Yeh, and Schneider, 1983) have examined the utility of time compression as an aid for training a basic air traffic control skill (a high performance skill). One group practiced intercepting an aircraft with the target plane traveling at 260 knots. The second group practiced the intercept at 5200 knots - 20 times real time! The subjects in the in the 260 knot group received 5-6 trials per hour during training, while those in the 5200 knott group received between 72-80 trials per hour. Both groups were then tested in real time.

The time compressed group was significantly better at identifying the turn point (i.e., the point at which the air traffic controller commands a turn in order to intercept an aircraft); there was no difference between groups on estimating roll out heading for the intercept.

Researchers at ECC and the University of Central Florida (Guckenberger, et al 1992) used a table top tank gunnery simulator to train subjects on three tank gunnery scenarios under five acceleration factors (i.e., 1.0x, 1.5x, 2.0x, sequential, and mixed). The results of this study demonstrated that training time could be cut up to 50%, with performance staying equal to (sequential, 1.5x) or surpassing (mixed, 2.0x) a real-time control group (1.0x). Further, in one ARTT group (mixed presentation) the mean performance score were 50% higher than the control group (1.0X). Another study (Guckenberger, et al 1993), used ARTT on F-16 part-task simulators and produced a 28% increase in the accuracy of performing emergency procedures by USAF pilots.

RESEARCH OBJECTIVES AND HYPOTHESES

The objectives of this study was to conduct research regarding:

1. The relative effectiveness of virtual time training versus conventional training in the same virtual environment. Specifically, this study attempts to systematically measure the benefits of Above Real-Time Training to subjects when they are transferred to real time testing conditions in a Virtual Reality environment.

2. The time adaptability of humans, that is, changes in time compression conditions of virtual time are mirrored by human time adaptability so that subjects are unable to differentiate between different time acceleration conditions. Specifically, this study attempts to measure the perceived workload demands of individuals in VT and real-time settings using well established and validated methods.

Based on prior research (Vidulich, Yeh, and Schneider, 1983, Guckenberger, et al, 1992, 1993), it was expected that training in a time accelerated environment would lead to poor performance versus a control group during training, but would lead to greater performance on a real-time transfer task . Second, based on the post test comments in prior ARTT studies (Guckenberger, et al, 1992, 1993), it was expected that VT subjects would be unable to differentiate between varied time conditions. Finally, it was expected that training under various time manipulations would not lead to negative transfer of training to a real-time task.

METHOD

Subjects

Twenty-eight university students served as subjects for this experiment. All subjects were recruited on a voluntary basis in accordance with American Psychological Association (APA) Principles for Research with Human Subjects. Prior to testing subjects were given written instructions informing them as to the general nature of the experiment.

Equipment and Materials

The experiment was run on a Virtual Environment testbed developed at the Institute for Simulation and Training and funded by ARI (Army Research Institute). The test-bed incorporates two 486-50 PC's with Intel DVI2 video cards, a Polhemus Fastrak with three sensors installed, a Virtual Research Helmet Mounted Display (HMD), a custom designed rapid gesture recognition glove (ChordGloves*), and a drafting table. The software was developed using the WorldToolKit library from Sense8 Corporation.

The Fastrak source was mounted and centered in the back of the drafting table. Two Fastrak sensors were used for viewpoint and right hand tracking. The viewpoint sensor was mounted on the front of the HMD with tip offsets adjusted to report values exactly at the center of the eyes. The hand sensor was mounted on the top of the ChordGlove and tip offsets were calibrated to report values at the point where the thumb and forefinger touch in a pinching gesture.

All viewing parameters were carefully calibrated to insure a one to one mapping between the drafting table in the real world and a model of the table in the virtual world. Standard WorldToolKit functions for parallax, convergence and Fastrak sensors were modified to improve this mapping. Sensor position and orientation from the hand sensor was directly coupled to a jack shaped cursor and acted as a 3D mouse.

Pinching contacts between the fingers and thumb, detected by the ChordGloves caused a cursor color change and, represented mouse button events.

Procedure

In this application of Virtual Time, twenty-eight subjects performed a simple tracking and targeting task under two levels of time compression, (i.e., 1.0x, 1.7x). All subjects were then tested in a real-time (1.0x) environment.

The individual subjects were asked to take a seat in a chair which had no wheels and move close enough to the drafting table so they could comfortably reach the top and center of the table. Subjects were given verbal instructions on what they were to look for in the virtual world. Subjects were informed that a 3D block would be moving back, forward left to right up and down while moving in a random 3D pattern. They were then instructed to place a glove onto their hand and a helmet was placed on their heads. Subjects were then told that the screen cursor represented a point between the forefinger and thumb. If they positioned the center of the crosshairs inside the target and pinched their thumb and forefinger together the target would disappear and end that trial. The subject's objective was to grab the virtual block as quickly as possible and each trial did not end until subjects successfully grabbed the block.

Each subject performed eighteen trials: Three familiarization trials, ten training trials and five testing (transfer of training) trials. Five subjects were randomly assigned to the Above Real-Time Training group (1.7x) and five to the control group (1.0x). Subjects were given the three familiarization trials at their assigned speed and then a one minute break. Next the ten training trials began, again at the same assigned speed. When this was complete, another one minute break was given. For the last five testing (for transfer of training) trials the control group was again tested at 1.0x, while the Above Real-Time VT group, who received training at 1.7x, was also tested at the 1.0x rate.

In order to determine if perceived workload demands were significantly different between the Above Real-Time VT group and the control groups three questionnaires were administered. These questionnaires were developed by modifying the Wewerinke (Wewerinke, 1974), which is a modified Copper-Harper scale (Cooper and Harper, 1969), and NASA Task Load Index (TLX) (Hart and Staveland, 1988) surveys. Both of these scales are well established and validated.

The modified Wewerinke scale (Wewerinke, 1974) was used as the basis for two questionnaires, one measuring perceived temporal demands and the other perceived mental demands. The temporal demand survey measured perceived temporal demands of the task on a scale ranging from 0- completely leisurely, very slow pace (i.e., an elderly person strolling through a park) to 9- frantic (i.e., the Olympic 100 meter dash). The mental demand survey measured perceived mental demands of the task on a scale ranging from 0- completely undemanding, very relaxed and comfortable (i.e., chewing gum) to 9- completely demanding (i.e., a time-pressured physics exam).

The modified NASA TLX scale (Hart and Staveland, 1988) was used to determine if there were any perceived differences in the following factors: mental, physical, and temporal demands, personal performance, frustration level, and effort level. The scale used to measure each of these factors ranged from Very Low (0) to Very High (100).

RESULTS

Raw performance data was collected after every trial. Summary data was then analyzed using a statistical T test. No significant difference was detected between the performance of the VT and control groups during both the training and testing phases. This is suggested to be due to the small sample size used for the study (n=28) and large variance in subjects initial VR skills. The trends in the data do, however, seem to indicate benefits to the VT group during testing. The mean of the 1.7x virtual time group (X=0.81 seconds, SD=0.73) was approximately fourty percent faster that of the control group (X=1.36 seconds, SD=1.42) during the testing phase (see Figure 1). The VT trend seems to indicate increased performance similar to prior ARTT studies (Guckenberger, et al 1992 1993).

This promising trend suggests further investigation is warranted.

The mean scores for both the real-time (1.0x) and the Virtual Time (1.7x) groups are tabled and graphically depicted on the following page.

Table 1 below depicts the means of both groups through all three phases of training (in Sec.).

	FAMILIARIZATION	TRAINING	TESTING
1.0x RT	3.78	1.94	1.36
1.7x VT	5.57	3.27	0.81

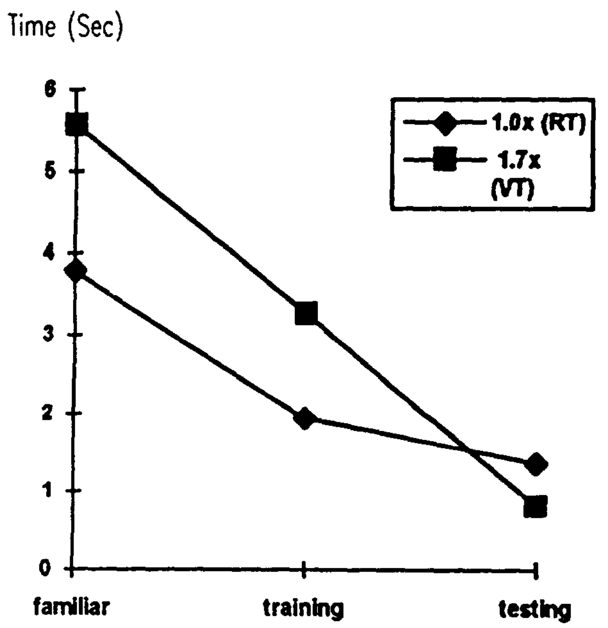


Figure 1: Descriptive statistics for the VT and Control groups through all three phases of training

Although the experiment did not detect any significant performance differences between the Above Real-Time VT and Control groups (which as aforementioned may be due to the power of the test) it would be beneficial to examine if there were any perceived workload differences between these two groups.

There were no significant differences detected in the perceived mental demands of the Above Real-Time VT group (training phase: X=3.8, SD=1.10; testing phase: X=3.3, SD=1.20) and the Control (training phase: X=3.7, SD=2.68; testing phase: X=4.0, SD=2.98,) groups (training phase : t=0.077, testing : t=0.487, neither of which are significant at the 0.1 level) using the modified Wewerinke scale.

For the training phase there were no significant differences detected in perceived temporal demands between the Above Real-Time VT group (X=4.6, SD=2.51) and Control (X=4.6, SD=1.14) groups (t=0) using the modified Wewerinke scale. For the testing phase, however, there was a significant difference in perceived temporal demands between the two subject groups. The Above Real-Time VT group (X=3.8, SD=1.10) perceived significantly less temporal demands that the Control group (X=5.6, SD=1.52) during the testing phase (t=2.15, which is significant at the 0.1 level).

The results form the modified NASA TLX scale indicated that the only factor for which a significant difference was detected between the two groups was frustration level. The Control group (X=20, SD=19.04) perceived significantly less frustration that the Above Real-Time VT group (X=54, SD=27.93) during the training phase (t=2.249, which is significant at the 0.1 level)

These survey results indicate that the Above Real-Time VT group, by receiving training at above-real-time rates, tended to find testing at real-time rates less time pressured than the Control group, who received training at real-time rates. Whether this perceived difference in temporal demands translates into differences in performance has yet to be fully verified. The results do indicate, however, that the above-real-time training rates tend to elicit a higher level of frustration than real-time training rates.

These results suggest that the subjects were unable to distinguish when they were in 1.0x from the 1.7x Virtual Time environments. The questionnaires results thus support the hypothesis that subjects would be unable to differentiate between different time acceleration conditions.

CONCLUSIONS

Based upon the results of this pilot study, tasks that contain simple psychomotor components such as the virtual block grab task seem to benefit form virtual time training, at least in terms of a reduction in perceived temporal demands. The small sample size used in the study is suggested to be of insufficient resolution to show statistical significance in performance time, but the trends seem favorable (see Figure 1) and bear future investigation with larger population samples.

1) It was hypothesized that virtual time training would be more effective than conventional training in the same virtual environment. This study did not detect a significant difference in performance time between the two groups, but did show the benefit of a significant reduction in perceived temporal demands for the VT group as compared to the control group during testing. In addition, there were no differences detected in the perceived level of temporal demands within each subject group. This result validates the prior ARTT study post test comments regarding the inability of subjects to differentiate between varied time conditions.

2) It is interesting to note, that both subject groups (1.0x & 1.7x) verbally complained and accused the experiment administrator of "speeding up the blocks", and "making the test harder" after their first one minute rest period between familiarization and training. The time rate was constant for both groups going from familiarization to training! It is suggested that the one minute rest period in virtual "blank" space, with its lack of active environmental stimuli, slowed down a subject's time norm. It is further proposed that when subjects transitioned back into the virtual training environment, their time norm, which had been recalibrated to the "blank" state, was disturbed thus leading to a higher level of perceived temporal demands. This anecdotal evidence suggests that the transition time to readjust the time norm in this case was one (1) minute or less. Although the subjects' comments have no scientific weight, it bears remembering that the original ARTT application success was in response to anecdotal comments from NASA test pilots. The transition time between different time norms is thus of interest and should be a target of future research efforts.

Finally, as expected training under various time manipulations did not lead to any negative transfer of training to a real-time task (i.e., the VT group did not perform significantly slower than the control group during the testing phase). As aforementioned, this was expected due to the similarity in the task stimuli and response requirements of the training and testing phases (Holding, 1965).

A key finding was the significant differences in the perceived temporal demands between subject groups. The VT group perceived less temporal demands during the testing (1.0x) phase than the control group. These results indicate that VT is a potential means of exploiting an existing ability of humans (time adaptability) within virtual training environments in order to achieve performance enhancement in real-time situations. Virtual Time as applied to the intrinsic time adaptability of man is a vast new field of great potential.

It is worth noting that adding VT to an existing Virtual Reality environment for this experiment was a low cost software only change with the software modification requiring less that 6 man hours. The low implementation cost and large potential benefits coupled with current economic conditions suggest VT as a timely solution.

The research results from this experiment support the on going synthesis of ARTT or Fast-Time Simulation into a cohesive theory. The current theory schematic (i.e. Appendix A) below encapsulates the progression and evolution of ARTT theory into Virtual Time.

Future Research Directions

Near-term work will focus on expanding the application of VT for emergency procedure training. The Silicon Graphics FLIGHT, DOGFIGHT and Shadow simulations have been successfully altered to support faster than real time rates. Virtual environment versions of these simulations are already being developed on a variety of manufactures hardware and will be altered to show Virtual Time in flight simulation.

The overall aim of the VT concept is to exploit the time adaptability of humans and foster a new way of thinking about time manipulation in the human-machine interface. Future research directions include safety, education, medical, and entertainment applications. For example, it would be possible to increase the voice and data communication rate over a virtual network to allow crews or teams to train at faster that real-time rates. Time flow could be manipulated for the benefit of the trainee. New training methods that are *time flexible* would change the form, fit and function of the human-machine interface. Above Real-Time VT programs are initially planned in simulation and training with follow on efforts involving the use of VT in human-machine interfaces.

Emergency procedure training for pilots, both commercial and military is envisioned as the initial proving ground.

The use of Virtual Time to investigate the nature and limits to the time adaptability of man.

Research the transition time to change "time norm settings", investigate if the transition time is dependent on the difference in environmental time rate changes, or a fixed transition time unaffected by the amount of environmental time rate change.

VR emersive nature binds human's time norm even more tightly than conventional simulation.

REFERENCES

Cooper, G.E. and Harper, R.P. (1969). The use of pilot ratings in the evaluation of aircraft handling characteristics. Washington, D.C.: *NASA Report* No. TN-D-5153.

Fraisse, P. (1984). Perception and estimation of time. In M.R. Rosenzweig & L.W. Porter (Eds). *Annual Review of Psychology 35* 1-36.

Guckenberger, D., Ullano, K.C., & Lane, N.E (1992). *The application of above real-time training for simulators: Acquiring high performance skills.* Paper presented at the 14th Interservice/Industry Training Systems and Education Conference, San Antonio, TX.

Guckenberger, D., Ullano, K.C., & Lane, N.E. (1993). Training High Performance Skills Using Above Real-Time Training. *NASA Technical Report Contract NAG-2-750.*

Hart, S.G. and Staveland, L.E. (1988). *Development of NASA-TLX (Task Load Index)*: results of empirical and theoretical research. Amsterdam, Netherlands: North-Holland.

Hoey , R. G., (1976). *Time compression as a means for improving the value of training simulators. Unpublished manuscript.*

Hoffman, R.G., & Morrison, J.E. (1987). *Requirements for a device-based training and testing program for M1 gunnery. Volume 1: Rationale and summary of results* (Report No. FR-TRD-87-41). Alexandria, VA: Human Resources Research Organization.

Holding, D.H. (1965). Principles in training. Oxford, England: Pergamon.

Jones, Mari R. Time Our Lost Dimension *Psychology Review 1976* Vol. 83 (5) 323-355

Hahn J. & Jones M. J. (1981) Invariants in Auditory Frequency Relations, Scandinavian Journal of Psychology 22, 129-144.

Kolf, J. (1973). *Documentation of a simulator study of an altered time base* Unpublished manuscript.

Lane, N.E. (1987). *Skill acquisition rates and patterns: Issues and training implications* New York: Springer-Verlag.

Lee, T.D., & Magill, R.A. (1983). The locus of contextual interference in motor skill acquisition. *Journal of Experimental Psychology: Learning, Memory, and Cognition 9* 730-746.

Matin, E., & Boff, K.R. (1988). Information transfer rate with serial and simultaneous visual display formats. *Human Factors* 30, 171-180.

Parasuraman, R. (1986). Vigilance, monitoring, and search. In K.R. Boff, L. Kaufman, & J.P. Thomas (Eds.). *Handbook of perception and human performance: Volume II: Cognitive processes and performance* (pp. 43-1 - 43-39). New York: Wiley.

Schneider, W. (1985). Training high performance skills: Fallacies and guidelines. *Human Factors 25* 285-300.

Shea, J.F., & Morgan, R.L. (1979). Contextual interference effects on the acquisition, retention and transfer of a motor skill. *Journal of Experimental Psychology: Human Learning and Memory 5* 179-187.

Shiffrin, R.M., & Schneider, W. (1977). Controlled and automatic human information processing: II. Perceptual learning, automatic attending, and a general theory. *Psychological Review, 84* 127-190.

Skelly, June J. (1993) *The Role of Event Time in Attending* Time & Society Vol. 2(1) . 107-128

SPSS, Inc. (1992). *SPSS for Windows* [Computer Program]. Chicago, IL: SPSS, Inc.

Stevens, S.S. (1975). *Psychophysics: Introduction to its perceptual, neural, and social prospects* New York: Wiley.

Thompson, M. (1965). *General review of piloting problems encountered during simulation flights of the X-15* Paper presented at the 9th Annual Meeting of the Society of Experimental Tests Pilots.

Toumodge, S.S. (1990) *Signal and Data Processing of Small Targets* Paper presented at the Orlando April 1990 Annual Meeting of the Society of Photo-Optical Instrumentation Engineers. (A91 - 36901 15-32) 378-385.

Vidulich, M., Yeh, Y.Y., & Schneider, W. (1983). *Time compressed components for air intercept control skills* Proceedings of the 27th meeting of the Human Factors Society. 161-164.

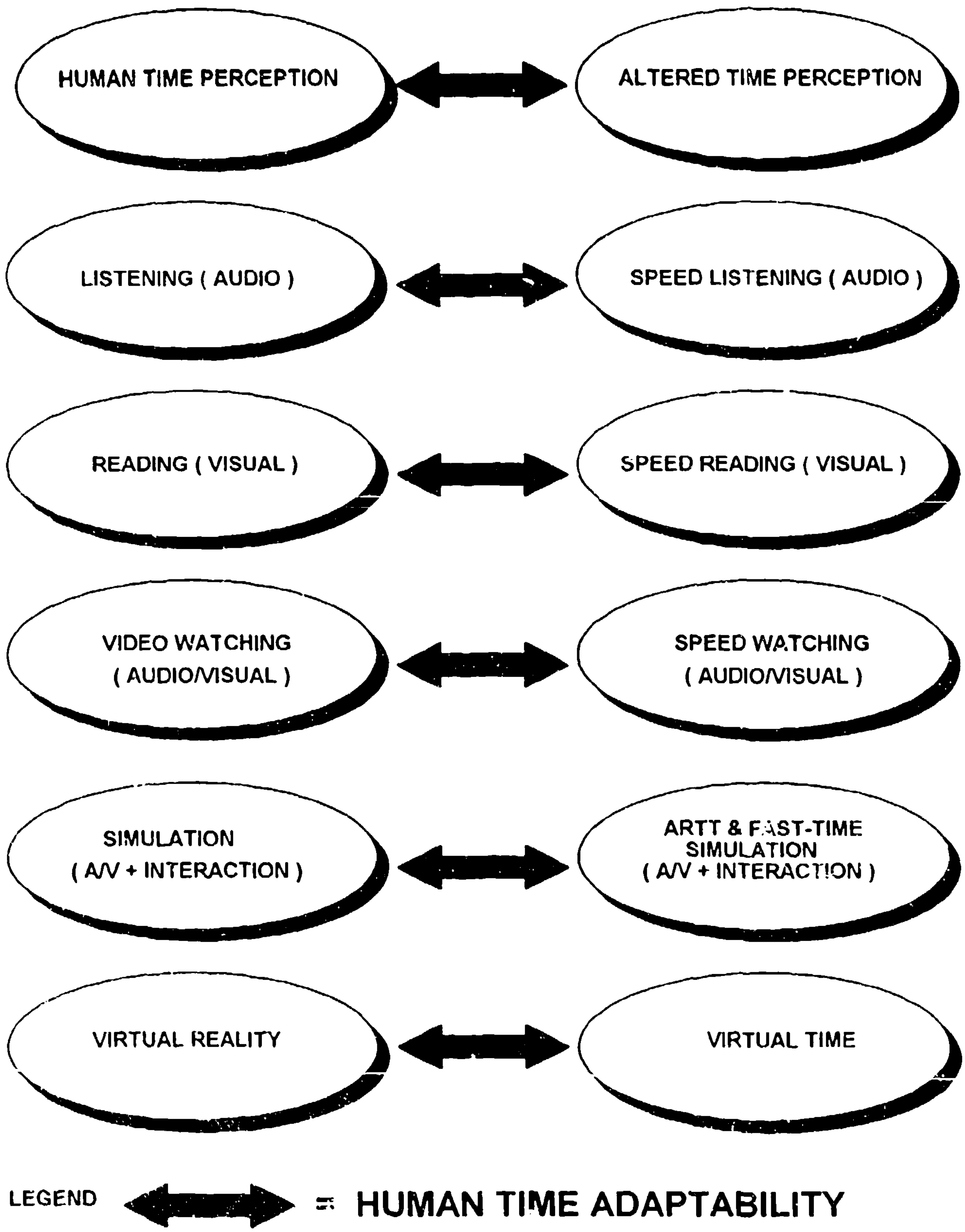
Wewerinke, P.H. (1974). Human operator workload for various control conditions. Wright-Patterson Air Force Base, OH, 10th NASA Annual Conference on Manual Control, 167-192.

ACKNOWLEDGMENTS

This research effort is an extension of the original research which was made possible by a grant from the NASA Dryden Flight Research Center (DFRC). Special thanks to Vern Carter of Northrup for his kind assistance in tracing the history of "fast time" simulation back to Jack Kolf at NASA DFRC. Mr. Kolf was the first to originate the concept of "fast time" simulation while working on the NASA X-15 project. His encouragement and support of our research effort began long before the actual support contract. His prior experience and suggestions were invaluable. We would also like to thank NASA's Dryden's chief test pilot Rodgers Smith and USAF 412th Test Wing's chief test pilot Dr. Pete Demitry for their support and suggestions.

The authors would like to acknowledge ECC and UCF for supporting our research with time, equipment, and facilities.

APPENDIX A Theory Schematic



A CONTEXT-BASED REPRESENTATION OF TACTICAL KNOWLEDGE FOR USE IN SIMULATION-BASED AUTONOMOUS INTELLIGENT PLATFORMS

Avelino J. Gonzalez
University of Central Florida
Orlando, FL, 32816

Robert H. Ahlers
Naval Training Systems Center
Orlando, FL 32826

ABSTRACT

The focus of the investigation described in this paper is the development of a concise, yet rich knowledge representation paradigm that could be effectively and efficiently used to model the intelligent behavior of simulated agents in a simulator-based tactical trainer. The behavior of these agents would be similar to that of an adversary who would react to a student's action in a manner representative of enemy tactics. The availability of this feature would be of significant utility to the training process for two reasons: 1) the student would face a realistic enemy who is knowledgeable about tactics in the domain of interest and, 2) the instructor would not have to be burdened with playing the part of the enemy in those training systems where this is commonly done.

The hypothesis presented is that whereas tactical knowledge is highly dependent upon the context (i.e., the situation being faced), a combination of script-like structures and pattern-matching rules in an object-oriented environment could serve as a concise means of representing the knowledge involved, as well as an efficient means of reasoning with that knowledge. This hypothesis was tested through the development of a prototype system that implemented the knowledge of a submarine tactical officer on a patrol mission. The prototype was implemented in CLIPS 5.1, a rule and object-based expert system shell developed by NASA. The results of the prototype show that the combination of scripts and rules in an object-oriented environment promises to meet the requirements described above.

ABOUT THE AUTHORS

Dr. Gonzalez is an Associate Professor of Electrical and Computer Engineering at the University of Central Florida in Orlando. His main area of interest is the application of artificial intelligence to the problems of simulation and training. He spent the Summer of 1992 at the Naval Training Systems Center as an ASEE Fellow, where the majority of the work described in this paper was developed.

Dr. Robert Ahlers is a Research Psychologist with the Human/Systems Integration Division of the Naval Training Systems Center. He has managed research projects concerned with the application of knowledge-based modeling to the simulation of intelligent agents within a training environment. His recent work addresses the use of artificial intelligence techniques to provide certain instructional features, such as performance diagnosis and feedback, instruction sequencing, and tactical adversary capabilities, within an embedded training environment.

A CONTEXT-BASED REPRESENTATION OF TACTICAL KNOWLEDGE FOR USE IN SIMULATION-BASED AUTONOMOUS INTELLIGENT PLATFORMS

Avelino J. Gonzalez
University of Central Florida
Orlando, FL, 32816

Robert H. Ahlers
Naval Training Systems Center
Orlando, FL 32826

1.0 INTRODUCTION

The use of intelligent simulated agents in a training simulation can provide a more realistic training experience to the students than would be presented otherwise. This is especially true for tactical trainers where an intelligent agent representing an adversary is able to react and counter the student's action in a realistic fashion.

In many training simulators, the instructor typically controls the simulated adversaries, providing them with intelligent behavior. But this can be quite burdensome to the instructor. The presence of intelligent agents in a training simulation that can autonomously react to the student's action in a realistic fashion would not only serve to increase the effectiveness of the training process, but also make it more efficient by off-loading some of the tasks from the instructor. Intelligent simulated agents will be henceforth referred to as Autonomous Intelligent Platforms (or AIP's).

But creating truly intelligent AIP's is a difficult task due to the many potential variations of any scenario, and the multiple actions that can be taken for each scenario. A rule-based paradigm appears to be a natural way to represent this knowledge, but the numerous conditions resulting from the many variations can translate into a large number of rules for even relatively simple tasks. Alternative representation and reasoning paradigms such as model-based, constraint-based, or case-based reasoning, although promising in some respects, (see Borning[1] and Castillo[2]) are not "natural" for this knowledge.

Thus, a significant obstacle to the full-scale development of AIP's for training simulators is the lack of a concise, yet rich, representation paradigm and reasoning scheme for the knowledge and behavior involved in such a task. Before describing the approach, however, a look at the type of knowledge to be represented may be warranted.

1.1 Tactical Knowledge

Tactical knowledge is typically general in nature. An example of general tactical knowledge is:

> *When facing an inferior enemy who has a tactical disadvantage, attack as soon as possible with overwhelming force.*

While the rule is seemingly simple and straightforward, it may be quite difficult for a non-expert to identify what constitutes an "inferior enemy" or a "tactical disadvantage". It is, therefore, not the knowledge of the above axiom that separates an expert from a non-expert, but rather, the ability to recognize an inferior enemy when one is faced, and a tactical disadvantage when pertinent conditions are analyzed. This can be described as the ability to correctly recognize the situation and place a tactical decision in its proper context.

An expert would look for key features such as the enemy's numbers, their heavy weaponry, their defenses, the surrounding terrain, the weather and the enemy's ability to fight in it, the element of surprise, etc. These would allow him

to identify the enemy's strength and evaluate his advantage or disadvantage.

Tactical experts are proficient at their task by recognizing and treating only the key features of the situation, and abstracting these for use as the premises for the general knowledge. They only use a small portion of the available inputs, but they know which ones to use.

An example of this is the tactical exercise of driving an automobile. A driver is generally bombarded with a multitude of inputs when driving: audio inputs such as engine noise, road noise, traffic noise, a blaring radio, conversation with passengers, etc.; visual inputs such as the instruments, other automobiles, the surrounding scenery, pedestrians, etc.; tactile inputs such as vibrations of the car, the position of the steering wheel, the gear shifter, the clutch, etc. These inputs are handled almost subconsciously by an expert driver when they are all in the normal or expected range. However, if one of these should become abnormal, such as the noise and vibrations that result from a tire blowout, the expert driver will immediately focus on these in order to recognize the present situation as a blowout, while ignoring all the other ones. This is referred to as *situational awareness*.

A beginning driver, on the other hand, may require significantly more concentration on the many inputs perceived, and his/her ability to classify the situation may be slower, and/or less accurate. In the case of a blowout at highway speeds, he/she may know what course of action to follow when faced with a blowout, but may not recognize in a timely fashion that the sounds and vibrations felt and heard are indicative of a blowout.

Most tactical actions in the military consist of a pre-defined set of actions which are embarked upon after a certain situation has been recognized. The situation could be a mission, a set of orders, or merely a reflection of a specific set of battle conditions at the moment. The problem faced by military tacticians, therefore, is two-fold: 1) how to recognize the present situation (situational awareness), and 2) what to do when the situation is recognized (referred to as *actionable information*).

1.2 Description of the Approach

The approach described here to address the above problem is based on the following hypotheses:

1) There is only a limited number of things that can take place in any situation. Using the example of the automobile driver, it would not be normally expected that a tire blowout take place while waiting at a stop light. This can be used to advantage to prune the search space of the problem, since there is no need to consider a blowout while waiting at a stoplight. Getting rear-ended, on the other hand, is a much more likely proposition.

2) The presence of a new situation will generally alter the present course of action to some degree. For example, the recognition of a blowout at highway speeds will cause the driver to attempt to coast to a stop while maintaining a firm grip on the steering wheel, and directing the car towards the shoulder of the road. Thus, the context changed from one of "normal driving", to one of "blowout", with its attendant actionable information. This context remains in effect until the car comes to a complete stop, at which point another situation will be recognized and acted upon (e.g., get out of car, inspect tire, change tire).

By associating the potential situations and corresponding actions to specific contexts, the identification of a situation can be simplified because only a subset of all possible situations is applicable under the active context. This context-based approach also easily addresses what actionable information to use when a situation is recognized.

One approach to implementing the approach described above lies partly in the use of a script-like concept. A *script* is a knowledge representation paradigm developed by Schank[3] which attempts to capture the actions, objects, persons, and concepts that may be related within a given context. For example, a restaurant script will be composed of all the actions which are typically part of going to a restaurant, such as reading the menu, ordering

the meal, eating it, paying the bill, etc. A restaurant script also contains props, objects which are typical to a restaurant scene from the customer's standpoint, (e.g., tables, chairs, menus, food, eating utensils, napkins, salad bars) as well as actors (e.g., waiters, hostesses, chefs, busboys). The actions involved are only those typical of the restaurant experience. It would not be normally expected, therefore, that the customer wash his car at the restaurant.

This concept can be easily extended to military tactics. A script can be used in this application to express the set of steps (at either a high or low level) that are necessary to carry out the action required by the present situation. Within the context of a mission, there is a limited number of things that are generally expected in terms of actions to carry out and the expectations in regards to the possible situations. It would be quite difficult to represent all this knowledge using rules alone. Thus, the basis for the work described here is the use of a script-like structure combined with a minimal number of rules as the knowledge representation paradigm for a set of AIP's. For lack of a better name, this representation and reasoning paradigm will be referred to as a *context-based representation.* The next section describes in greater detail how context-based representation can be implemented to achieve the objectives generally set for AIP's in a simulation.

2.0 GENERAL DESCRIPTION OF CONTEXT-BASED REPRESENTATION PARADIGM

Scripts are used to represent specific contexts or situations, and will thus, be referred to as *contexts.* A context can be likened to a situation that has been recognized, and which has a prescribed set of procedures that must be carried out. Additionally, any situation, by its very nature, will limit the number of other situations that can take place. The behavior of the objects in the simulation are controlled by the context that is active at the time. Exactly one context must be active at any one time. A context is composed of:

- message-handlers that initialize the appropriate objects in the simulation at the point of initial activation of the context.

- message-handlers that execute certain actions during the time which the context is active.

- rules which are applicable only when the context to which they belong is active. These rules assess the situation, prescribe action to be taken within the context, or determine when a transition to another context is called for.

2.1 Control of AIP Through Contexts

The AIP is controlled by a *General Context (GC),* a high level context that defines the overall mission to be undertaken. The GC is an instance of a class representing the mission to be executed and it is composed of *Acts,* intermediate-level contexts that define certain maneuvers, situations, or tactics relevant to that mission. Acts can be "installed" in a sequential nature, or in response to a developing situation. The Acts, in turn, can have *Sub-acts* which are contexts describing lower-level tactics or maneuvers to be undertaken within the Acts.

Each context (GC, Act, or Sub-act) will have a set of rules and/or procedures attached which will implement some action and/or detect when a transition to another context is called for. The rules, in particular, form the basis of situational awareness. If a change in the parameters of the simulation calls for a change in the situation, the rules will recognize the change and execute the appropriate change of context. The use of contexts, in addition to prescribing actionable information as a block of procedures, can help in the situational awareness process by limiting the types of situations that can be expected. For example, in a context of peacetime and within territorial waters, a submarine would not expect to be attacked by an enemy.

Explaining the concept of context-based reasoning would be easier if the explanation is tied to an example. Therefore, the representation of submarine tactical knowledge during a routine patrol mission will be used as

an illustration. It is also the basis for the prototype described in section 3.0 below.

The AIP which is the recipient of the tactical knowledge will be referred to as *ownsub.* While this might cause confusion with the traditional naval custom of using this nomenclature to identify the student's own platform, this AIP <u>is</u> the focus of this investigation and should be recognized as such.

Ownsub is represented as an object (instance of class SUBMARINE) in an object-oriented environment. Its static slots (defined as those slots whose values will not change during the simulation) define its capabilities such as its maximum speed, quiet speed, maximum depth, periscope depth, weapon systems, (e.g., number and ranges of torpedoes, missiles), sensors (e.g., range and types of passive sonar, active sonar, radar, towed arrays), and Electronic Warfare capabilities (e.g., sonar decoys). Additionally, ownsub's dynamic slots (those whose values will be updated at least once during the simulation) describe its actual position (i.e., x-coordinate and y-coordinate), depth, heading, and speed, in the course of the simulation, as well as whether the sensing equipment is on or off. Damage assessment as well as the conditions of the stores (weapons, food supply, etc.) is also contained in dynamic slots.

A local database containing all of the external information that is relevant to the mission is also necessary. This includes all mission-dependent static inputs which define the task to be undertaken, as well as the environment that will be experienced by ownsub. Some of these are the:

1) mission,

2) geographical info.,

3) presence of friendly forces in the sector or in the route,

4) basic assumptions about the state of affairs (e.g., peacetime, tensions, war), as well as others, such as coordination with other friendly forces.

This local database should also contain all the situation-dependent dynamic inputs which ownsub would see from its sensors. Such information partially consists of:

1) sonar input,

2) radar input,

3) communications with the command or with other friendly ships or submarines.

This type of information should be placed on the database by a central "manager" function that can access all the data in the simulation, yet knows what data is to be made visible to ownsub. It is recommended that a blackboard architecture with hierarchically-arranged blackboards be used to implement this feature. (This, however, was not employed in the prototype.)

The general description of a context (without the inputs) could be stored in memory or on disk. A context can be "installed" onto an AIP object as may be called for by the situation, overwriting the old one when the change represents mutually-exclusive contexts. If the new context is not mutually-exclusive with the existing one, then it needs to be overlaid onto the old context so that at its completion the original one regains control.

Each context contains procedural attachments to implement procedural control over the simulation, such as dictating the speed, depth and bearing of ownsub. Moreover, the context would also contain a set of rules together with its own "mini" inference engine consisting of a pattern matcher, a Rete net and an agenda, as well as the capability to assert and retract facts from the local factbase, to call procedures, and to change contexts.

One context is always in control of the situation. This is indicated by a fact asserted into the factbase which identifies the context in charge. For example:

(general-context SEARCH-AND-TRACK)

General Contexts are, by definition, always mutually exclusive with one another. If a change

of GC is indicated by the external inputs (i.e., new orders from fleet commander), or by internal reasoning about the situation at hand, then the new GC replaces the old one. The fact that "advertises" the old GC is removed from the factbase and one indicating the new one is asserted.

The Acts of a GC are generally mutually-exclusive with one another, but each co-exists with its parent GC. When a particular context (Act or Sub-act) is in effect, it is considered to be the *active-context*. For example:

(active-context SECTOR-SEARCH)

Only one context can be active at any time, other than the GC, which is active as long as it is valid. If the GC is replaced, however, its active-context must also be deactivated. When a mutually-exclusive context (Act or Sub-act) is to be installed, the presently active context is deactivated and considered to be the *previous-context*.

(previous-context TRANSIT-TO-SECTOR)

This introduces a rudimentary ability to reason temporally when it is important to know what ownsub was doing previously.

Sub-Acts are treated as Acts, except they are not necessarily considered to be mutually-exclusive. They are simply overlaid on top of the active Act. If a non-mutually-exclusive context is overlaid on an active context, the presently-active context is considered to be the *background-context*

(background-context TRANSIT-HOME)

while the new one assumes the role of active context. Upon deactivation of the latter, the background context is re-installed as active if it is still valid. The contexts themselves contain the knowledge of which other context they are or are not compatible with.

The actions to be taken as prescribed by a particular context is done by sending messages to the appropriate objects. For example, if the SECTOR-SEARCH Act becomes active, then a message is sent to ownsub which sets its speed equal to the quiet speed, turns off the active radar, etc., in order to quietly search the sector. Each context has one initialization message that is sent to ownsub to initialize the actions of the AIP.

2.2 Situational Awareness and the Setting of Contexts

The above section described the procedure which governs how the appropriate contexts are installed and allowed to control the AIP. This section will describe how it is determined that a context is no longer valid and must be replaced or temporarily over-ridden. This deals with the situational awareness, or threat assessment issue of context-based reasoning.

The basic recognition of the situation is done through pattern-matching rules. While this might not sem to be a concise way of carrying this out, the use of the active-context in the rule premise will significantly limit the solution space of the search as was described in the previous section. Rules will have a pattern in their premises that indicates the active-context to which they are applicable. Only when there is a fact in the factbase indicating the active status of the appropriate context will these rules be "active" and capable of being executed.

There are basically two types of rules involved in the intelligent decision-making: *Sentinel rules* continually monitor the simulation data in order to recognize the factors that can lead to a change in situation. These rules are typically tied into a particular context. Sentinel rules form the basis of situational awareness, since it is these that infer the new situation, and therefore the context, from raw data.

Transition rules, on the other hand, react to the situation identified by the sentinel rules and determine which context should be activated as a response to the changing situation.

An example of a sentinel rule is one where, when the TRANSIT-TO-SECTOR (full-speed travel to the sector to be patrolled) context is active, a rule belonging to that context will monitor the position of ownsub so that arrival at the designated sector is recognized. This rule requires the fact

(active-context TRANSIT-TO-SECTOR)

be present in the factbase. The action of this rule may designate the situation to be

(situation arrived-in-sector ownsub).

A transition rule will recognize this posted fact, and react by activating the SECTOR-SEARCH context, and initializing all appropriate elements. Once the new context is activated, the sentinel rule mentioned above is no longer applicable, since its related context has been deactivated.

Some sentinel rules, however, will always be "active" regardless of which General-Context/Act/Sub-act is active. These are general rules that oversee the entire simulation and are not tied to any one context. For example, a rule that searches for enemy torpedoes needs to be always in an active status whenever ownsub is out of port in a wartime context, whether there are enemy submarines detected or not. If an under-attack situation is identified, then a transition rule will immediately activate the UNDER-ATTACK context. Such rules do not require any active-context facts to be present in the factbase in order to execute.

Please note that the above section generally describes how the context-based representation should be implemented. It does not, however, describe exactly how it was actually implemented in the prototype described in Section 3.0 of this report. The prototype represents some simplifications of the description contained above.

3.0 IMPLEMENTATION AND VERIFICATION OF CONTEXT-BASED PARADIGM

The context-based paradigm was implemented in a prototype in order to verify that 1) it can be used to suitably represent tactical knowledge, and that 2) it can do so concisely. The more specific objective of the prototype was to implement a simple mission of searching a pre-determined sector for the presence of enemy submarines, and to track one when found. Such a mission was labeled *SEARCH-AND-TRACK*. The knowledge implemented in the prototype is described in detail in Gonzalez[4].

Object-oriented languages excel in applications to simulations. Objects can be defined and assigned a certain behavior, which can be controlled by sending messages to it. Since the prototype is, in its most basic terms, a simulation, an object-oriented language was chosen as the implementation tool. Since the endowment of intelligence to the object ownsub was the primary goal of the prototype, an expert system tool which could manipulate rules was also desirable. Many commercially-available expert system shells incorporate these features. CLIPS 5.1 was chosen for the prototype due to its powerful pattern-matching capabilities, its newly-available Clips Object-Oriented Language (COOL), its procedural capability, its availability in a PC platform, and its low cost.

The basis of this prototype is the instantiation of the class SUBMARINE, called "ownsub", which represents the AIP. Ownsub traverses a Cartesian coordinate plane of unlimited size in three dimensions (x-coordinate, y-coordinate and depth). It's actions are controlled by a set of contexts that are based on the General Context SEARCH-AND-TRACK. The Acts that compose the SEARCH-AND-TRACK GC are called:

- TRANSIT-TO-SECTOR
- SECTOR-SEARCH
- COVERT-TRACKING
- TRANSIT-HOME

These four Acts are mutually-exclusive, and are described in detail in Gonzalez[4]. Whenever any one of them is installed, a message is sent to ownsub to initialize its parameters for operation under this context. This involves setting its speed, heading and depth, activating or deactivating certain sensors, deploying its weapons, etc. A context is installed by asserting a fact to the factbase that advertises it as the active context. These facts were discussed in Section 2.0 above, and are of the form:

(active-context <context>)

The prototype is centered around the actions of ownsub. Objects representing up to 5 enemy submarines can be created, and they are referred to as *opsub1*, *opsub2*, etc., through *opsub5*. Additionally, other objects in the

simulation can be created to represent torpedoes (up to 3 at one time) and sonar decoys (up to 5). Only the ownsub and opsub$n$ objects contained any attributes that changed during the course of the simulation.

The performance objective of the prototype was to indirectly control the actions of ownsub by directly controlling those of opsub. The only exceptions to these was when specific orders were given to ownsub.

Situational awareness can be thought of as the reasoning necessary in order determine when to effect a transition to another context. The sentinel rules described in section 2.0 form the basis of this process. For example, the situation where an enemy is detected is determined by a calculation of the distance between the positions of ownsub and that of all existing opsubs. If the active sonars of all submarines are off, then ownsub's passive sonar will detect the presence of the enemy at 3000 meters. If ownsub's active sonar is on, then that distance increases to 4000 meters. Likewise, if the enemy's active sonar is on, not only is the range then 5000 meters, but it is assumed that the enemy is aware of ownsub's presence, something that was not assumed in the previous two cases. Losing track of an enemy happens when the distance is greater than 500 meters above the applicable range. The range of the torpedoes is roughly 7 Km, which is implemented as a run duration of four minutes at a speed of 100 Km/hr.

It should be noted here that while the above is admittedly a mis-representation of the capabilities of submarines, their weapons, and their sensing equipment, (the ranges and speeds are known to be much different than this), the relatively slow speeds of the submarines involved require that short distances be used in order to limit the duration of the simulation to a reasonable one, while at the same time being able to exhibit the many features of the prototype.

A sentinel rule also searches for the presence of torpedoes, and recognizes them as such when the objects representing torpedoes come within the appropriate range as described above. Other sentinel rules monitor simulation for the end of an enemy attack, check to see whether a torpedo has hit the target, and when to end an evasion maneuver, among many other things.

In some cases, rules were written such that they skipped the intermediate step of explicitly asserting the situation and integrated the functions of the sentinel and the transition rules into one. While this has the effect of making the system more concise, which was one of the objectives of this prototype, it tends to de-modularize the knowledge, something that could be disadvantageous when carrying out the knowledge engineering for a significantly larger system.

4.0 EVALUATION AND DISCUSSION OF CONTEXT-BASED PARADIGM AND PROTOTYPE

The purpose of this section is to describe the findings that were made during the process of developing the prototype, discuss the lessons learned, and mention where enhancements to the context-based representation could be made as a result of further research. It is of particular interest to analyze the models described above for their conciseness, since that is considered a critical issue.

The issue of concise representation is a critical one which merits careful evaluation. This section begins by looking at the CLIPS elements used in the prototype and examines their nature. One way to measure conciseness is to simply count all the CLIPS elements used in the prototype. However, a difference must be made between elements used for the purpose of carrying out the simulation, versus those used for intelligent decision-making. Therefore, all the elements for each of the prototypes will be counted, but only those used for decision-making will be used in the analysis of conciseness. The prototype used the following types of CLIPS elements: classes, global variables, functions (time-related, general functions, and main functions), message-handlers, and rules.

The classes represent SUBMARINE, SECTOR, GENERAL-CONTEXT, SEARCH-AND-TRACK, TORPEDO, and SONAR-DECOY. Of the global variables defined, 10 were used to represent time points, while the rest represent ranges (of

torpedoes, sonar-decoys, etc.). Of the 13 general functions, only 11 were used for intelligent decision-making.

Of the 27 message handlers, 9 were used as part of the simulation (to move the objects in their prescribed direction.) Thus, only 18 were used for decision-making.

In the rules, 16 of the total of 50 were specifically used for the process of asserting and retracting facts in the factbase which describe data contained within the objects (e.g., position, active-context's). These can be considered to be for purposes other than decision-making. Thus, only 34 were used for decision-making. The list below shows the final tally of elements used by the prototype for intelligent decision-making.

- 6 classes
- 15 global variables
- 13 time-related functions
- 11 general functions
- 2 main functions, (initialization, and cycling)
- 18 message-handlers
- 34 rules

Ideally, it would be of great benefit to develop a purely rule-based version of the knowledge and capabilities exhibited by the prototype, so that an objective comparison of efficiency and conciseness could be made. Nevertheless, a rough comparison can be made with another situational knowledge-based system prototype developed at the first author's institution. The latter prototype is a performance monitoring system that evaluates the actions of a student in an automobile driving simulator. Both systems are similar in that they represent situational awareness knowledge and both were developed with CLIPS, but the performance monitoring prototype represents its knowledge in rules alone. While the submarine AIP prototype employing context-based reasoning required 34 rules and 18 message handlers, the performance monitor required nearly 50 rules and numerous other functions to perform a much simpler situational recognition mission. This comparison, rough as it may be, supports the hypothesis that the context-based paradigm is capable of representing the knowledge involved in the SEARCH-AND-TRACK mission in a concise manner. It is also likely that additional refinements in the system could lead to an even more succinct representation.

5.0 SUMMARY AND RECOMMENDATIONS

This research attempted to determine whether the use of a combination of script-like structures called contexts, objects and rules would represent a concise, yet rich paradigm for modeling AIP's. The results of the research indicate that the context-based paradigm can be used to concisely represent the knowledge required of an AIP in a training simulation. Nevertheless, additional work remains in order to make the concept a "user-hardened" technology.

Most importantly, a formalization of the context-based reasoning concept is necessary. The resulting formal knowledge representation paradigm should incorporate features that address temporal reasoning as well as dealing with uncertainties. Additionally, guidelines should be developed to assist a user in creating a knowledge base from human tactical knowledge. These guidelines could be incorporated in a highly interactive graphical authoring system with a look and feel similar to that of the VISTA system[5]. Lastly, the burden of carrying out the simulation should be placed on a more generalized simulation environment, with the CLIPS prototype merely providing the intelligence and the control of the simulated agents. This would make the system more widely applicable and possibly capable of being retrofitted to existing simulators.

Furthermore, additional testing in a more complex threat environment is clearly warranted so as to confirm or deny its general applicability to the problem. Lastly, an implementation in an existing training simulator should be undertaken to determine the feasibility of retrofitting the latter with AIP's.

6.0 ACKNOWLEDGEMENTS

The investigation described in this paper was carried out by the author under the sponsorship of the Naval Training Systems Center, Human Factors Division, Orlando, FL, and under the auspices of the ASEE Summer Faculty Program, in the Summer of 1992.

7.0 REFERENCES

1. Borning, A., "ThingLab - An Object-oriented System for Building Simulations Using Constraints", *ACM Transactions on Programming, Languages and Systems*, 3 (4) October, 1977, pp 353 - 387.

2. Castillo, D., "Toward a Paradigm for Modelling and Simulating Intelligent Agents", Doctoral Dissertation, University of Central Florida, December, 1991.

3. Schank, R. C., and Abelson, R. P., 1977, *Scripts, Plans, Goals and Understanding*, Erlbaum, Hillsdale, NJ, 1977.

4. Gonzalez, A. J., "Concise Representation of Autonomous Intelligent Platforms in a Simulation through the Use of Scripts", Technical Report TR92-019, Naval Training Systems Center, Orlando, FL, 1992.

5. Ahlers, R. and Schnitzius, M., "Human Engineering the Knowledge Acquisition Interface: The Evolution of VISTA", *Proceedings of the The Third Annual Florida Artificial Intelligence Research Symposium*, Cocoa Beach, FL, April, 1990.

6. Khboshch, Vladimir Alekseyevich, "Submarine Tactics", Voyenizdat, 1989, translation in 1990, by the Foreign Broadcast information Service.

7. Labayle-Couhat, J. and Prezelin, B., *Combat Fleets of the World 1988 89*, Translated by A.D. Baker III, Naval Institute Press, Annapolis, MD 1988

EVALUATING THE OVERHEAD OF OSI STACKS IN INTEROPERABLE DIS NETWORKS

Margaret Loper- Institute for Simulation and Training
M. A. Bassiouni- Department of Computer Science
G. Bulumulle- Institute for Simulation and Training
Ming Chiu- Department of Computer Science

University of Central Florida
Orlando, Florida

ABSTRACT

OSI (Open Systems Interconnections) communication stacks can be used to interconnect heterogeneous DIS machines and eliminate their incompatibilities. However, the interoperability benefit of OSI stacks could be offset by the computational overhead associated with the complex data transformation process of OSI upper layers. It is feared that an OSI implementation utilizing the transformation process would be too slow to meet the real-time requirements of DIS networks. In this paper, we present the results and conclusions of a detailed performance evaluation study which we have recently conducted to measure the overhead of the OSI transformation process, assess its impact on the delay encountered by DIS PDUs, and evaluate the benefits of using lightweight transfer syntax implementations.

ABOUT THE AUTHORS

Margaret Loper received her masters degree in Computer Engineering from the University of Central Florida in 1991. She is currently a Systems Engineer for the Institute for Simulation and Training, Orlando. Ms. Loper has been actively involved in research on simulation networking, OSI protocols, and multicast communications. She is an active member of the DIS Communication Architecture and Security Subgroup and is also a member of ANSI working groups on OSI Upper Layers and Network & Transport Layers.

M. A. Bassiouni received his Ph.D. degree in Computer Science from the Pennsylvania State University in 1982. He is currently an Associate Professor of Computer Science at the University of Central Florida, Orlando. He has been actively involved in research on computer networks, distributed systems, concurrency control, and relational databases. Dr. Bassiouni is a member of IEEE and the IEEE Computer and Communications Societies, the Association for Computing Machinery, and the American Society for information Science.

G. Bulumulle received his masters degree in Computer Engineering from the University of Central Florida in 1992. He is currently an Associate Engineer at the Institute for Simulation and Training and is involved in research related to DIS networks.

M. Chiu received his masters degree in Computer Science from the University of Central Florida in 1991. He is currently a Ph.D. student and Research Assistant in the Department of Computer Science, UCF.

EVALUATING THE OVERHEAD OF OSI STACKS IN INTEROPERABLE DIS NETWORKS

M. Loper, M. Bassiouni, G. Bulumulle, and M. Chiu

University of Central Florida
Orlando, Florida

INTRODUCTION

In large scale distributed interactive simulation (DIS) systems[6,10], various heterogeneous computing nodes are used as vehicle simulators and as control and data logging elements. Today, there are two great standards shaping the architectural principles and the technology of connecting large number of heterogeneous computing machines, namely, the OSI (Open Systems Interconnections) reference model and the Internet protocol suite (colloquially known as TCP/IP). The OSI model uses the principle of layered architecture in a more rigorous way, but the efficiency of the initial OSI implementations has been traditionally lower than that of TCP/IP. Both standards have similar lower-level communication (Physical and Data Link) layers that can employ Ethernet, token ring, FDDI, and other LAN/WAN protocols. Notable differences exist in the Transport layer of the two standards (i.e., the OSI transport protocol and TCP). Some of the functions of the Transport layer include: segmentation and reassembly of user messages, routing and flow control, recovery from data loss, and congestion avoidance. The OSI model defines three distinct layers above the Transport layer. These are the Session layer (which organizes the structure of the message exchanges), the Presentation layer (which allows a mutually acceptable transfer syntax to be established between the communicating entities), and the Application layer. The Internet protocol stack does not have this upper layer structure; rather, the functions of the Sessions and Presentation layers are built into the Application layer as needed. The placement of the implementation of the transfer syntax as well as the common abstract language upon which it is based (e.g., ASN.1, XDR, Xerox's Courier) represent only one of the differences between OSI and Internet.

OSI-compliant communication stacks can be used to interconnect the heterogeneous DIS machines and eliminate their incompatibilities as will be explained shortly. OSI, or the corresponding Government mandate (GOSIP), is a network protocol architecture consisting of seven layers. The upper layers of OSI are the place to implement any common syntax for OSI-compliant networks. Specifically, the OSI Standards place the functionality of the transfer syntax in the Presentation layer of the OSI stack, and make it a selectable feature that is not required for compliance. OSI upper layers therefore may perform a complex transformation step which produces a common transfer syntax for the exchanged messages. The interoperability benefit of this aspect of the OSI stack is, however, offset by the computational overhead associated with producing and managing the common transfer syntax. It is feared that an OSI implementation utilizing the common transfer syntax for DIS networks would be too slow to meet the real-time requirements of interactive training and would therefore degrade the realism of the training exercise. The objective of this paper is to present the numerical results and conclusions of a detailed performance evaluation study which we have recently conducted to measure the overhead of the OSI transfer syntax, assess its impact on the delay encountered by DIS PDUs, and

evaluate the benefits of using lightweight transfer syntax implementations.

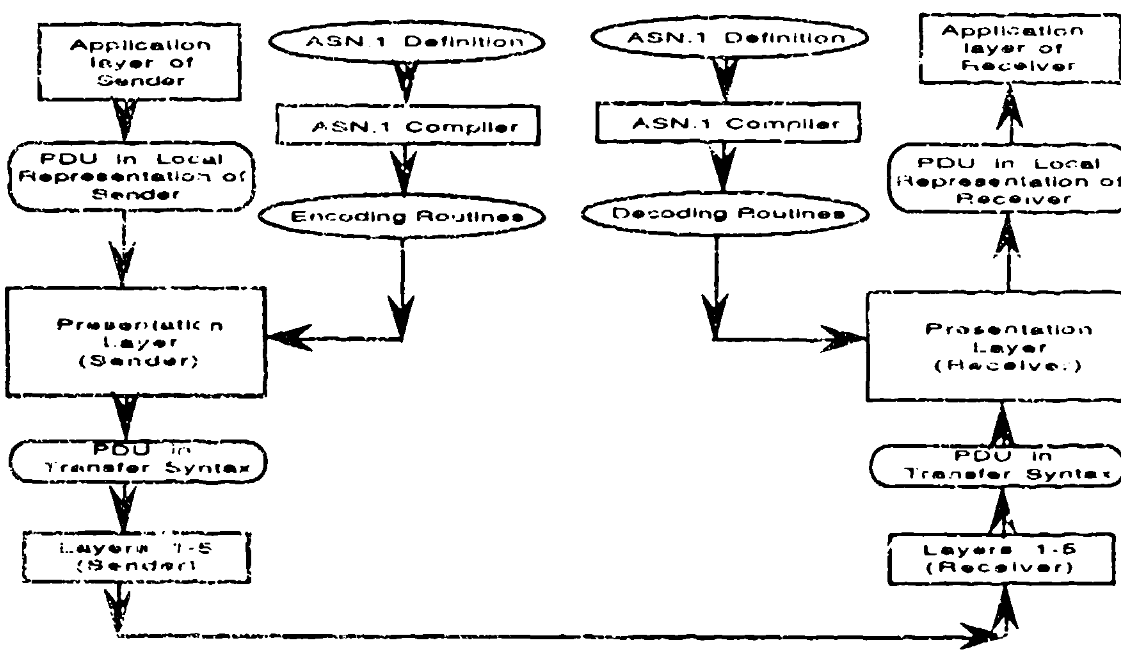
The performance experiments were conducted using the DIS/OSI Testbed at the Institute for Simulation and Training. The OSI stack was provided by the ISO Development Environment (ISODE)[7,8], a widely used suite of software primarily designed for fast implementation and testing of OSI upper-layer protocols. Using ten PDU types of the DIS Standards, our tests enabled us to evaluate the throughput delay encountered by a DIS PDU with and without the OSI transfer syntax overhead. The ten PDU types used in our tests, from DIS Version 1.0[10], are:

1) Entity State
2) Fire
3) Detonation
4) Collision
5) Service Request
6) Resupply Offer
7) Resupply Received
8) Resupply Cancel
9) Repair Complete
10) Repair Response

For each PDU type, several experiments were performed to compute various performance measures (e.g., the average value, the standard deviation, and the coefficient of variation of the end-to-end delay, the degradation ratio due to the overhead of the OSI Presentation layer, etc.). Consistent results have been observed when the tests were repeated using four different hardware platforms. In the following sections, we describe the OSI/DIS performance experiments and present the performance results and conclusions.

CHARACTERIZATION OF OSI/ASN.1 OVERHEAD

Using OSI/ASN.1, two dissimilar DIS nodes can exchange protocol data units (PDUs) as illustrated by the following diagram.



Basically, the exchange mechanism is carried out as follows:

1) The PDU is first transformed from its local representation at the sending host to a transfer syntax using a set of

transformation rules called the *encoding rules*.

2) The PDU in transfer syntax is transmitted down the communication stack of the sending host, and is delivered in the same transfer syntax to the receiving host.

3) The PDU in transfer syntax is transformed to the local representation of the receiving host using a set of rules called the *decoding rules*.

For OSI-compliant networks[3], two standards have been so far proposed and adopted by ISO/ANSI: i) The *Abstract Syntax Notation One* (ASN.1)[2,4,9] is used to solve the problem of heterogeneous local representations of data, and ii) ASN.1 *Basic Encoding Rules* (BER)[5] are a set of encoding rules used to produce a transfer syntax for the exchanged data based on ASN.1. Consider for example the communication of a simple PDU between two machines: the sending machine A encodes characters in ASCII and uses a 2's complement scheme for integers; the corresponding representations in the receiving machine B are EBCDIC and 1's complement. The PDU transmitted by the application layer of machine A has two fields: an integer of value -5, represented in 2's complement, and an octet string of value "USA", represented in ASCII. For each of the two fields, the BER code in the Presentation layer of machine A generates a sequence of three components: 1) unique tag, 2) length identifier, and 3) the value of the field represented in a common transfer syntax. These Tag-Length-Value (T-L-V) sequences are transmitted down the stack of machine A and ultimately received by the Presentation layer of machine B. The BER decoding routines in machine B uniquely decipher the T-L-V sequence of each field and then passes the value -5 in 1'complement and the string "USA" in EBCDIC to its application layer.

The following are the different types of time overhead incurred by the implementation of OSI/ASN.1.

a) Encoding Overhead (EO): which is the time needed for the execution of the BER encoding routines.

b) Sender Processing Overhead (SPO): which is the extra processing time (excluding the encoding overhead) in layers 1 through 6 due to the representation of data in transfer syntax.

c) Decoding Overhead (DO) and Receiver Processing Overhead (RPO): these are defined analogously to EO and SPO.

d) Total Time Overhead (TTO): which is the sum of the above components. In the DIS environment, TTO represents the extra end-to-end delay encountered by a DIS PDU when the OSI transfer syntax is introduced.

PERFORMANCE EVALUATION EXPERIMENTS/RESULTS

To assess the impact of using OSI in DIS communication networks, several experiments were conducted using ISODE[7,8]. The following is a high-level description of these experiments.

The Isolation Model

The purpose of this experiment is to compute the encoding and decoding overhead, EO and DO, associated with OSI in DIS simulators.

The Network Model

In this experiment, measurements are taken with respect to the end-to-end delay between two hosts and the total time overhead TTO is recorded.

Table 1 gives the average end-to-end delay in milliseconds for each DIS PDU type with and without the overhead of OSI/ASN.1. Identical Sparc machines were used both as sender and receiver. The column labeled "without transfer syntax" gives information about the end-to-end delay encountered by a PDU when it is transmitted between two hosts without

invoking the OSI/ASN.1 encoding or decoding routines (i.e., the PDU is treated like a single stream of binary data which is transmitted without transformation). Each entry in Table 1 was obtained by transmitting the same PDU sixty times and computing the average value of the end-to-end delay and the corresponding coefficient of variation, denoted C.o.V., which is obtained by dividing the value of the standard deviation over the average value. The 60 samples used in computing the average delay were found to be statistically sufficient for obtaining accurate results (care was taken to avoid sampling the initial few transmissions in which higher delay is observed due to the cost of connection set-up). We also define the degradation ratio as the ratio between the increase in the average delay due to OSI/ASN.1 and the original average delay (without OSI/ASN.1). Figure 1 shows the histogram of the degradation ratio for two different hardware configurations: the first configuration uses only Sparc machines and the second uses only Motorola machines.

Table 1. Impact of OSI transfer syntax on the average end-to-end delay

PDU Type	without transfer syntax		with transfer syntax	
	Avg.	C.o.V.	Avg	C.o.V.
1	8.193	0.022	24.395	0.022
2	7.609	0.016	18.815	0.033
3	8.178	0.021	25.078	0.017
4	7.533	0.037	15.023	0.038
5	7.454	0.051	13.634	0.031
6	7.385	0.026	13.831	0.229
7	7.472	0.025	18.274	0.029
8	7.345	0.029	10.154	0.012
9	7.357	0.027	10.642	0.035
10	7.370	0.032	10.649	0.029

LIGHTWEIGHT OSI IMPLEMENTATIONS

In this section, we present our ongoing analysis and evaluation of light-weight OSI implementations, e.g., the skinny enveloping scheme. The latter approach is based on limiting the functionality of the OSI transfer syntax implementation to what is needed by the application and eliminating unused features. In the Presentation and Session layers, the approach works by pre-coding invariant octet-sequences for outbound messages. At the receiving end, the inbound messages are matched against the invariant octet-sequences for direct access of relevant data. If the match fails, the expensive process of general parsing (e.g., ASN.1 parsing) of the incoming octets is performed. In the best scenario, the match would be all what is needed to handle the envelope carrying the wrapped data. The skinny stack doctrine does not provide any guidelines regarding which fields can be encoded as "invariant octet sequences" or which fields can be ignored; the choice is basically application dependent. In the DIS Entity State PDU, for example, one may consider the fields : "protocol version" and "exercise-id" as invariant values (since the same value is used throughout one training exercise). Some or all of the various padding fields, and other entity dependent fields (e.g., "country", "category", "domain", etc.) may be practically ignored since they are not needed in every ESPDU transmission. Our performance tests were also used to determine the level of improvement that can be achieved when all nodes of a DIS network use a skinny OSI implementation. For each PDU type, experiments were performed to determine the end-to-end delay for the skinny enveloping case as well as for the general parse (full stack) counterpart. Our tests were executed on four different "sending host/receiving host" hardware configurations denoted by S/S, M/M, S/M, and M/S where S stands for a Sparc machine and M stands for a Motorola machine. Figure 2 shows the values of the end-to-end delay (in milliseconds) for the ten DIS PDUs using the S/S configuration.

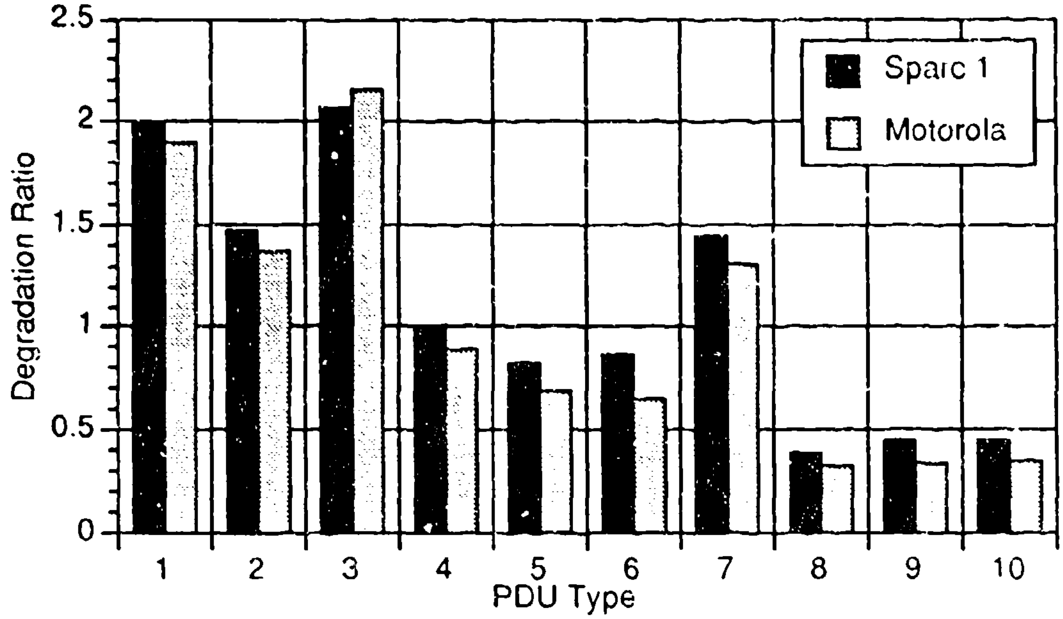


Figure 1. Degradation ratio for two configurations

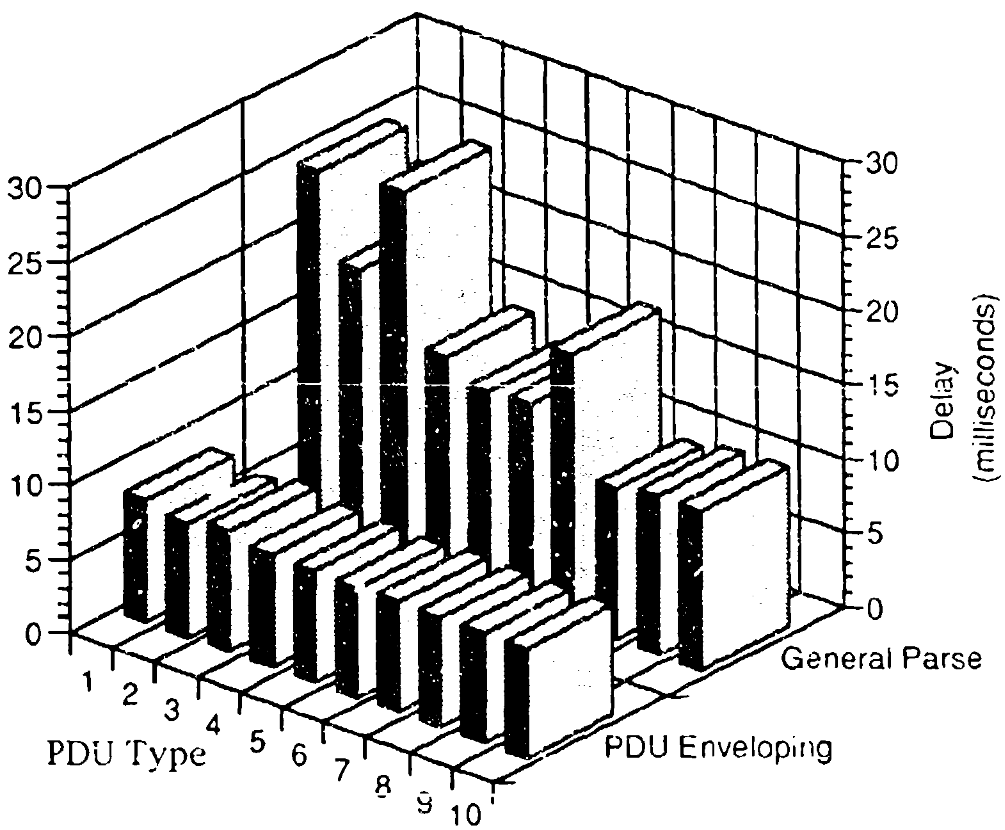


Figure 2. End-to-end delay for the S/S configuration

Figure 2 also shows that unlike the case of PDU enveloping of the skinny stack, the general parse of the full stack exhibits significant variations in the value of the end-to-end delay among the different PDUs. Furthermore, the speed-up of the skinny enveloping approach has been found to depend on the composition of the actual data transmitted. To help analyze the numerical results, and to better understand their implications, we shall introduce a simple metric that reflects the complexity of the different DIS PDUs. Table 2 shows the composition of these PDUs based on the description of their contents in the DIS Standards.

Table 2. Composition of PDU types in the DIS application

PDU Type	# integers	# real values	# octet strings
1	42	9	6
2	27	7	1
3	39	9	2
4	13	7	2
5	18	1	2
6	17	1	2
7	17	1	2
8	9	0	1
9	10	0	2
10	10	0	2

Let

I_k = the average number of integers in a PDU of type k

R_k = the average number of real values in a PDU of type k

S_k = the average number of octet strings in a PDU of type k

and define C_k to be a metric for the complexity of processing (e.g., parsing) a PDU of type k. A simple choice of C_k is the following linear relation:

$$C_k = a*I_k + b*R_k + c*S_k$$

where a, b, and c are constants. The bubble chart of Figure 3 shows the relationship between the speed-up and the complexity of the PDU for the Sparc hardware (the corresponding results for the Motorola hardware are quite similar and are not given in this paper). The chart contains a bubble for each PDU type such that the size of the bubble is proportional to the complexity metric of the corresponding PDU type (assuming a=1, b=4, and c= 4). The value of the vertical displacement (Y-axis) of the center of the bubble is equal to the speed-up achieved by the enveloping scheme. The speed-up is defined to be the ratio between the end-to-end delay of the DIS PDU using a full stack and the corresponding end-to-end delay using the skinny enveloping scheme. In general, the larger the size of the bubble, the higher the corresponding speed-up value. The choice a= 1, b=4, and c=4 in Fig. 3 to represent the complexity of integer, real, and string variables, respectively, was simply made based on the size we expected for these variables (many integers in DIS PDUs are short integers of size one byte; a real value is usually encoded in four bytes; and many string fields are used for padding and are of size four bytes). We have also experimented with other reasonable choices of a, b, and c. The results were not significantly different from those presented in the paper. Figure 4 shows the values of the speed-up for the four different hardware configurations. The minimum speed-up in Figure 4 has a value of 1.28 and corresponds to the Repair Complete PDU in the S/M configuration. The maximum speed-up has a value of 3.43 and corresponds to the Detonation PDU in the M/S configuration. Notice that the most frequent PDU in DIS (namely, the Entity State PDU) suffers from a very high ASN.1 overhead and would therefore benefit the most from lightweight transfer syntax implementations. Finally, it should be noted that the simple complexity metric derived from Table 2 didn't differentiate between integers and short integers and didn't take the length of individual octet strings into account. Using more sophisticated metrics is a topic that is worthy of further investigation.

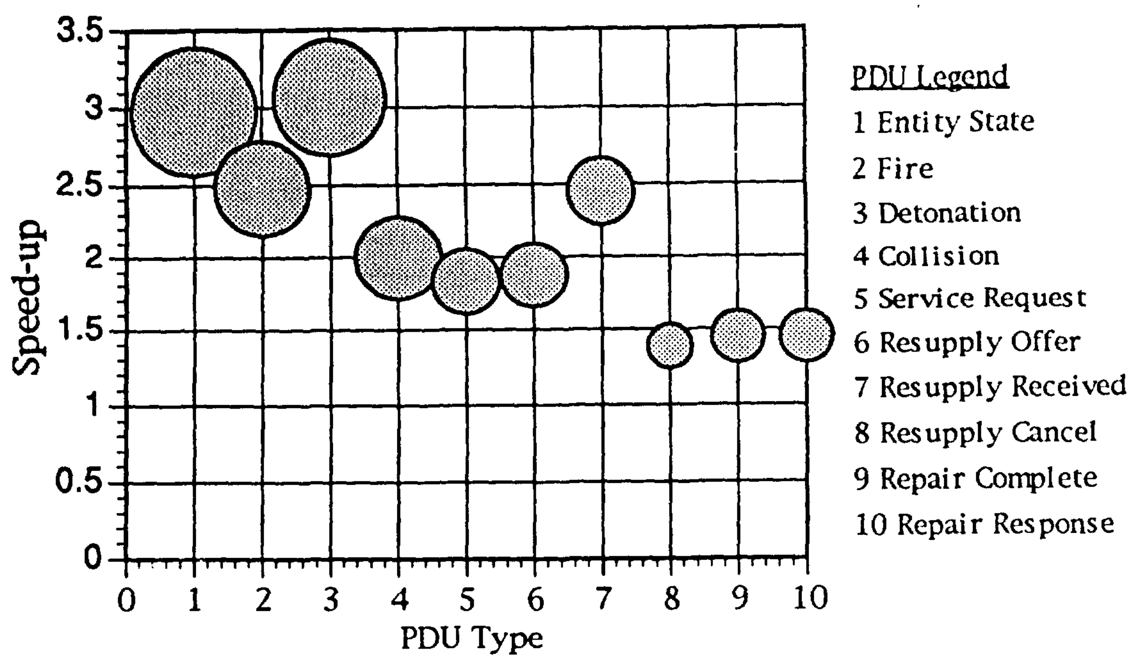


Fig. 3. PDU complexity vs. speed-up
(Note: bubble size is proportional to complexity)

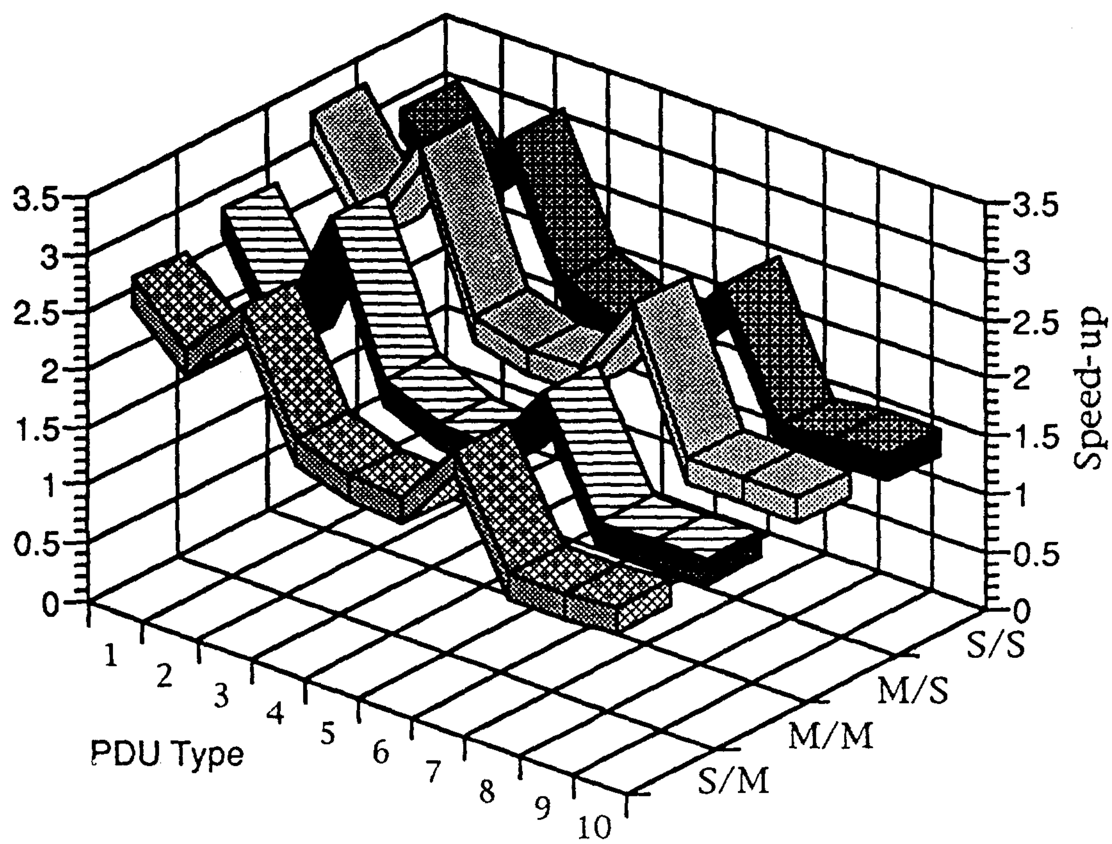


Fig. 4. Speed-up values for four hardware configurations

The range chart of Figure 5 gives the average speed-up value for each PDU (the average is computed over the four hardware configurations). The minimum and maximum values of the speed-up are also depicted.

In addition to the ten DIS PDUs discussed earlier, the Network Model was also used to measure the end-to-end delay of a PDU consisting of a sequence of m integers (as was done in previous work[11]). Figure 6 plots the relationship between m and the average end-to-end delay (in milliseconds using Sparc machines). The delay shown in Figure 6 is the overall delay encountered by a packet when traveling from one host to the other.

In general, the end-to-end delay was found to closely follow the linear equation

$$d = c_0 + c_1 * m$$

where c_0 and c_1 are constants, d is the delay in milliseconds, and m is the number of integers in the PDU ($c_0 = 7.582$ and $c_1 = 0.205$ for the Sparc hardware used in Figure 6). The corresponding delay, d', without invoking the OSI/ASN.1 routines can also be approximated by a linear equation

$$d' = c_0 + c_2 * m$$

where c_2 is a constant whose value is orders of magnitude smaller than that of c_1. A good approximation of the OSI/ASN.1 overhead TTO can therefore be obtained as follows

$$\begin{aligned} TTO &= d - d' \\ &\simeq c_1 * m \end{aligned}$$

In other words, the relationship of TTO versus m is similar to that shown in Figure 6, but shifted vertically by the value c_0.

CONCLUSIONS

In this paper, we presented the results of our performance evaluation experiments to measure the interoperability overhead of the OSI transfer syntax in DIS networks. The tests showed that the end-to-end overhead of OSI's ASN.1 is significant and can therefore compromise the proper operation of the DIS application. Our experiments also gave preliminary insight into the possible speed-up of the lightweight skinny approach. The tests showed a speed-up of up to 3.4. In most

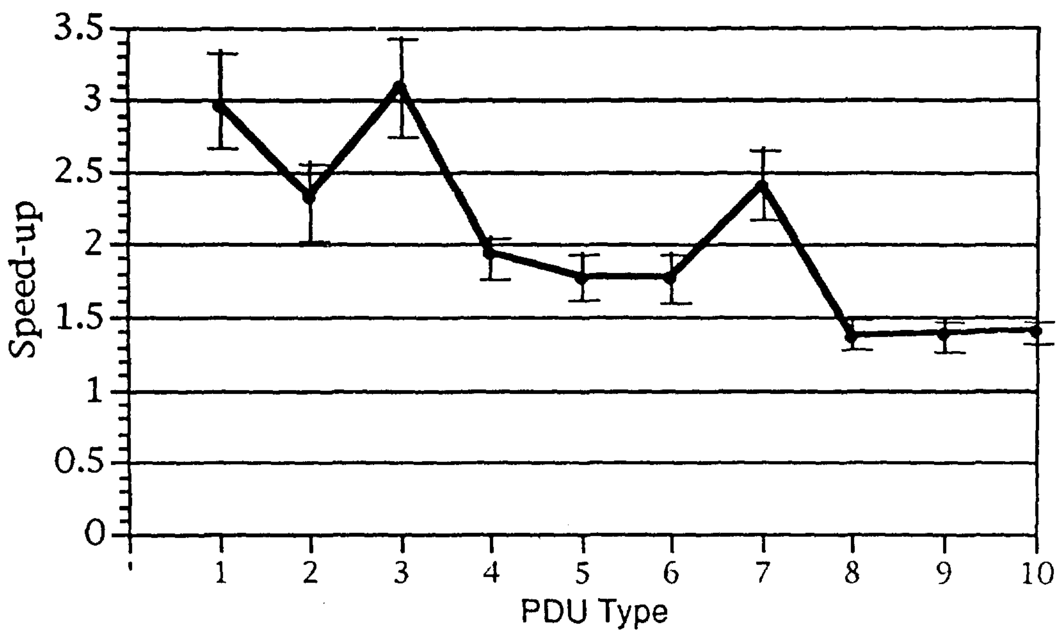


Fig. 5. Average value and range of speed-up

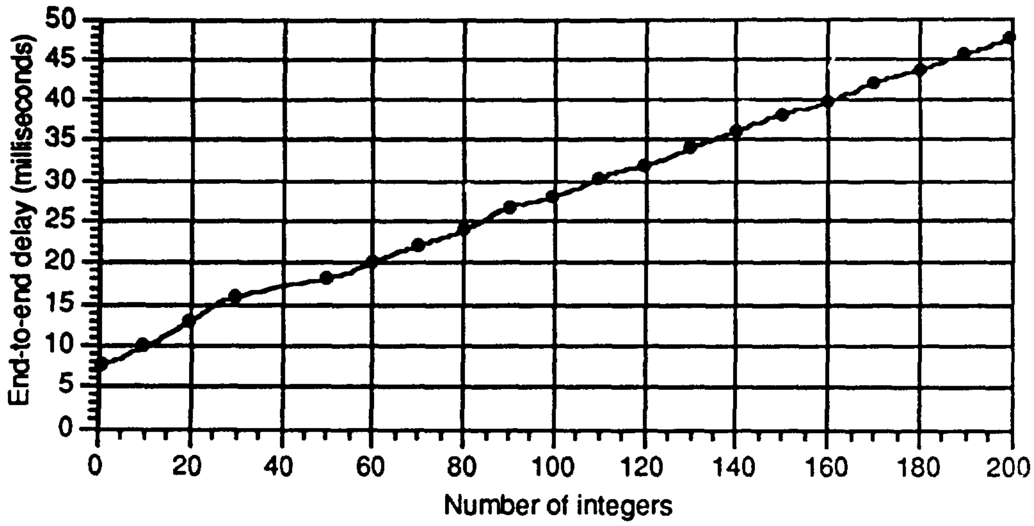


Figure 6. End-to-end delay for a PDU vs. number of integers

PDUs and hardware configurations, the speed-up is well below 3 implying that the skinny enveloping scheme in DIS is at most three times faster than the full stack. The minimum speed-up observed in our tests is 1.28. Although an actual skinny implementation for the DIS application may differ from the set-up used in our tests, the results reported in this paper clearly show that there is an evident need to develop and standardize lightweight OSI-compliant networks.

ACKNOWLEDGMENT

This work has been supported by U.S. Army's Simulation, Training, and Instrumentation Command (STRICOM) under contract N61339-91-C-0103.

REFERENCES

[1] Furniss, P. and Ashley, K. "The OSI skinny stack" Draft Report, University of London Computing Center, March 1992.

[2] Gaudette, P. "A tutorial on ASN.1" U.S. Department of Commerce, National Institute of Standards and Technology, Technical Report, NCSL/SNA-89/12, May 1989.

[3] ISO. Information Processing- Open systems Interconnection- Basic Reference Model. ISO International Standard 8072, 1983.

[4] ISO. Information Processing- Open systems Interconnection- Specification of Abstract Syntax Notation One (ASN.1). ISO International Standard 8824, 1990.

[5] ISO. Information Processing- Open systems Interconnection- Specification of Basic encoding Rules for ASN.1. ISO International Standard 8825, 1990.

[6] LORAL System Company. Strawman Distributed Interactive Simulation Architecture Document. Vol. I: Summary Description. ADST Program Office, March 1992.

[7] Rose, M. "ISODE: horizontal integration in networking" ConneXions, The Interoperability Report, Vol. 2, No. 3, March 1988.

[8] Rose, M. *The ISO Development Environment: User's Manual.* Volumes 1-5, Performance Systems International, Inc., January 1990.

[9] Steedman, D. *Abstract Syntax Notation One (ASN.1): The Tutorial and Reference.* Technology Appraisals Ltd. ISBN 1-871802-06-7, 1990.

[10] STRICOM: *Ditsributed Interactive Simulation- Standards Developement Draft 2.0*, September 1992.

[11] Neufeld, G. and Yang, Y. "The design and implementation of an ASN.1-C Compiler" IEEE Transactions on Software Engineering, Vol. 16, No. 10, 1990, pp. 1209-1220.

INFRARED ATTRIBUTES FOR PROJECT 2851 STANDARD SIMULATOR DATA BASES

Erwin Baumann, Budimir Zvolanek, and Douglas Dillard
McDonnell Douglas Training Systems
McDonnell Douglas Corporation
St. Louis, Missouri

ABSTRACT

The many phenomenological Infra-Red (IR) modeling programs currently in use require a large number of parameters to achieve a high degree of image simulation accuracy. When the parameter sets required by these programs are tabulated, the result is a large and diverse set of potential database attributes. In addition, these IR modeling programs are intended to satisfy the needs of a wide range of IR simulation users. To achieve the goals of Project 2851 in defining DoD Standard Simulator Data Bases, decisions must be made regarding which parameters should be included as attributes within the databases. The set of selected attributes must satisfy a wide variety of IR image simulation programs and users while being of reasonable size for storage in IR image generator databases.

McDonnell Douglas Training Systems assisted Project 2851 in the selection of these parameters by taking a three part approach to the task. First, current IR phenomenological models were studied and their required parameter sets were tabulated. Second, IR modeling experts and weapons systems users were surveyed to determine their needs. And third, a Quality Function Deployment analysis was performed to prioritize the parameters with respect to user needs, producing a set of IR database attributes that were recommended to Project 2851. This paper describes the results of the user survey, the evaluation process, and the recommended IR attribute set.

ABOUT THE AUTHORS

Erwin Baumann is a Senior Technical Specialist at McDonnell Douglas Training Systems. Mr. Baumann has specialized in the application of artificial intelligence methods to sensor simulation, machine vision, and automated planning. As a member of the Sensor Simulation and Database Technology group at MDTS, he has developed innovative applications of neural networks to multi-spectral image analysis and sensor simulation. Mr. Baumann received a B.S.E.E. degree from the University of Minnesota.

Budimir Zvolanek is a Principal Investigator in R&D at McDonnell Douglas Training Systems. Mr. Zvolanek has concentrated his career on the application and development of electronic imaging systems. He was program manager for Project 2851 database contracts at MDTS where he leads the sensor simulation and database product development activities. Mr. Zvolanek received his M.S.E.E. degree from Washington University in St. Louis, Missouri.

Douglas Dillard is a Lead Engineer at McDonnell Douglas Training Systems. Mr. Dillard has extensive experience in the design of visual and sensor simulations and the associated databases required by computer image generators used in aircrew training systems. He has made key contributions to Project 2851 and other simulation database research projects at MDTS. Mr. Dillard received a B.S. degree from St. Louis University in St. Louis, Missouri, and a M.S. degree from the University of Missouri at Rolla, Missouri.

INFRARED ATTRIBUTES FOR PROJECT 2851 STANDARD SIMULATOR DATA BASES

Erwin Baumann, Budimir Zvolanek, and Douglas Dillard
McDonnell Douglas Training Systems
McDonnell Douglas Corporation
St. Louis, Missouri

INTRODUCTION AND PURPOSE

Project 2851 (P2851) has incorporated data structures and attributes for Infra-Red (IR) sensor Image Generation within its Standard Simulator Data Base (SSDB). However, P2851 expressed a further need for an improved set of IR simulation attributes satisfying training, simulation, and weapon systems user requirements.

McDonnell Douglas Training Systems (MDTS) has been conducting on-going research and development efforts to improve the fidelity and rapid generation of simulated IR imagery. These efforts have included studies in environmental IR modelling and the definition of requirements imposed by training, mission rehearsal, and weapon systems users and experts.

As a result, MDTS was contracted by P2851 to recommend an improved set of IR simulation attributes based on IR sub-system training requirements, existing target and background IR imaging models, and IR simulation attributes currently defined by P2851. The objectives of the contract were to evaluate and suggest improvements to the IR simulation attributes for IR Generic Transformed Data Bases (GTDB) and SSDB Interchange Format (SIF) to support a high level of simulation fidelity with respect to dynamic time, weather, and atmosphere-dependent IR simulations. Since attributes in the GTDB and SIF are derived from the SSDB, any IR attribute recommendations would apply to the SSDB as well as GTDB and SIF.

To meet these objectives, a systematic IR attribute evaluation and selection process was performed. The steps taken were: 1) Identify and categorize training-dependent IR image attributes; 2) Survey IR image simulation models and selected database attributes; 3) Analyze IR simulation attribute relative importance; 4) Evaluate currently defined SIF and GTDB IR simulation attributes; and 5) Select and recommended a prioritized IR simulation attribute set for SSDB use.

IR SIMULATION USER-NEEDS SURVEY

To support a user-driven approach to defining required and desirable SSDB IR attributes, selected members of the IR simulation user community were asked to identify which characteristics of IR imagery are most important to them in various training contexts. These user-prioritized IR image attributes were then related to a large set of IR database attributes obtained from a study of current IR simulation models. These relationships were combined with the user- specified priorities using MDC Quality Function Deployment (QFD) methods to derive a set of prioritized database attributes capable of supporting the simulation of the desired IR image characteristics.

User Communities

Input from a broad spectrum of IR image simulation users was solicited. The user communities that were addressed included fighter pilots, Bombardier/ Navigators (B/Ns), the Automatic Target Recognition (ATR) community, and other IR simulation experts.

An evaluation form was prepared to allow users to specify the importance of various aspects of IR

imaging simulations. The survey form included sections on "Sensor Controls" and "Sensor Effects" in addition to "Background Characteristics" and "Target Characteristics". Recipients of the survey were asked to specify the importance of various IR image attributes in the context of three IR image simulation categories: Initial Training, Advanced Training, and Mission Rehearsal. The training categories can be described more completely as follows:

1) Initial Training' – Generally limited to a period of several weeks using highly scripted scenarios with objectives to:

- Learn operating procedures
  - Functions and applications of system features
  - Displays and controls
  - Handoffs, coordination, and timing of events
- Integrate Forward Looking Infra-Red (FLIR) operation, navigation, and weapons
  - Consolidate visual tasks (e.g., target acquisition) with systems tasks (e.g. weapons release)
  - Sequences of operations
  - Workload management

2) Advanced Training' – A broad category between "Initial Training" and "Mission Rehearsal" with objectives to:

- Upgrade skills to employ new systems or weapons
- Operate under simulated threats
- Develop and evaluate tactics for various scenarios

3) Mission Rehearsal – Defined here as flying a specific mission profile utilizing the full complement of on-board sensors, correlated with a highly geo-specific visual scene that displays a 3-D perspective environment in real-time. No avoidable disparity between the simulated and actual mission that would jeopardize the mission's success is permitted.

Users were specifically asked to provide input about IR image characteristics (e.g., "diurnal effects") rather than the underlying database feature attributes (e.g., "emissivity") from which the IR image could be generated. However, they were free to include their own suggestions for IR database attributes if they so desired.

Initial Training Needs

Analysis of the survey responses yielded the following observations regarding the training importance of various IR image characteristics.

First, it is clear that there is little consensus concerning the importance of specific IR image characteristics in most cases. That is, for nearly every image characteristic listed, someone believed it was very important and someone else believed that the same characteristic was minimally important. This was true of most background and target characteristics with the exception of "Accurate Relative (Background) Intensities" which was consistently considered to be "Very Important" to "Moderately Important".

Second, there was general agreement that simulation of sensor controls for initial training was very important. The simulation of sensor effects was consistently judged to be more important than background or target characteristics, but less important than sensor controls.

Perhaps the most useful overall observation to be made in the context of "Initial Training Needs" is that learning the sensor controls and observing basic sensor effects are more important than the accuracy of background and target intensities. This is not surprising since the geographic areas and targets being observed can often be generic for this level of training.

Advanced Training Needs

The observations made for the Initial Training category also apply here with the following modifications. First, although there was still no strong consensus concerning the importance of specific IR image characteristics, the number rated as "Minimally Important" dropped dramatically. There was general agreement on the importance of "Accurate Relative (Background) Intensities" with most users considering this aspect of an IR simulation to be "Very Important".

Mission Rehearsal Needs

Mission rehearsal needs represent (excluding ATR applications) the most demanding set of IR simulation requirements to be addressed by the SSDB. Observations made for the Advanced Training category also apply here but with increased

importance ratings for all image characteristics and sensor effects.

The accurate simulation of sensor controls remained of highest importance for mission rehearsal scenarios. Although the accurate simulation of IR targets increased in importance, accurate IR background simulations continue to be viewed as more important than either IR target appearance or sensor effects.

Another fact of particular significance is that "Diurnal Effects" and "Weather/Environmental Effects" are considered more important than any other background effects in a mission rehearsal scenario. This is expected as the training becomes increasingly focused on specific geographic areas, particular targets, and expected environmental conditions.

IR SIMULATION MODELS

A brief review of the methods and needs of current IR modelling systems may help the reader better understand the constraints leading to the final list of recommended IR attributes.

Conventional IR Modelling Methods

Conventional IR simulation models divide their complete predictive model either explicitly or implicitly into a "Physical (World) Model" component and a "Sensor Model" component as shown in Figure 1.

The "World Model", as depicted in Figure 2, calculates the amount of reflected, transmitted, and emitted IR energy present by solving equations based on conservation of energy principles and energy transfer rates constrained by the objects and environmental conditions. Inputs to the world model include static feature attributes such as reflectance and dynamic environmental attributes such as cloud cover. Dynamic environmental attributes including

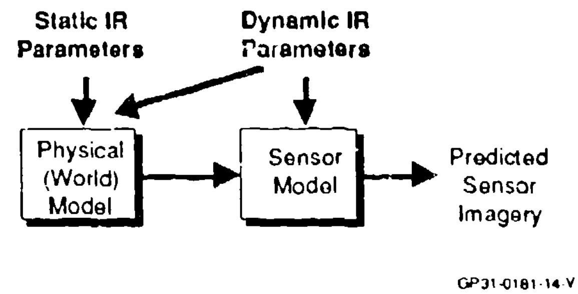


Figure 1. IR Model Components

humidity, haze, smoke, and dust levels are used to simulate atmospheric effects, primarily further attenuation of the input IR (generally solar) energy. Attenuation of the output IR energy on its way to the sensor can be predicted using the same dynamic IR attributes plus knowledge of the sensor's position.

The "Sensor Model" is responsible for determining what portion of that energy is in the sensor's field-of-view and adding sensor-specific effects such as noise, AC coupling (swathing/striping), and blooming. These sensor effects can be influenced by the settings of sensor controls including gain and level. Inputs to the sensor model are the output from the world model and any dynamic attributes that may affect the sensor's operation.

IR Model Attribute Survey

An extensive survey of existing IR image simulation models produced a comprehensive set of database attributes potentially useful for accurate, model-based, image generation.[2] As an initial step in the selection process, the database attributes for each model were separated into dynamic and static attribute sets.

Static attributes include properties such as thermal mass or emissivity which generally remain constant

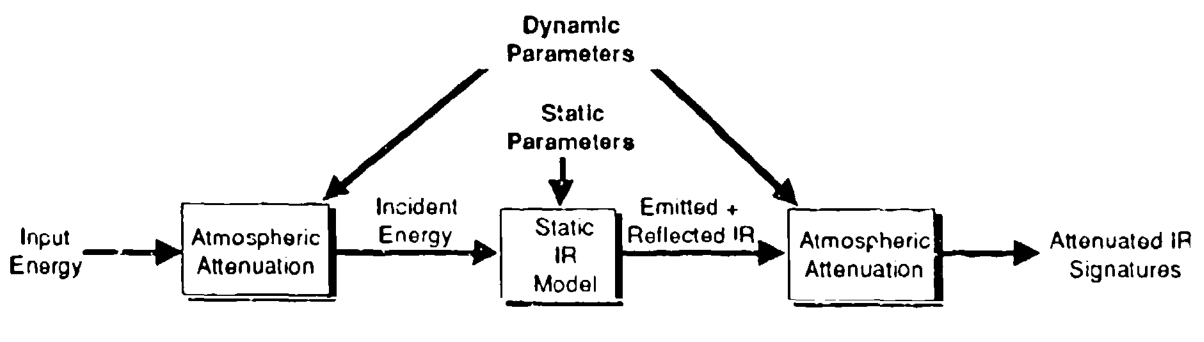


Figure 2. World Model

for an object and are natural feature and model descriptors to be included in the SSDB. Dynamic attributes such as cloud cover, air temperature, humidity, time-of-day, snow cover, haze/smoke/dust levels, and other environmental attributes are important inputs to IR sensor simulations, but are usually initialized by a user at simulation run-time and are therefore not appropriate for inclusion in the static IR attribute set of the SSDB.

This study focused on the static attribute set used to describe terrain and objects. The recommended database attributes set consists of the union of those IR attributes already in the SSDB with a selected subset of the large, comprehensive, static attributes set derived from the IR simulation model survey.[2]

RELATIVE IMPORTANCE OF IR ATTRIBUTES

The IR attribute set recommended here represents a necessary compromise among several competing factors including generality, redundancy, convenience of use, availability of values, and database size. These constraints were balanced with the overriding need to satisfy a variety of IR simulation users and models.

QFD Evaluation Method

Quality Function Deployment (QFD) methods were used to convert the user- specified IR image attribute rankings into a set of prioritized IR database attributes[3,4]. The format of the QFD matrix for IR database attribute evaluation is shown in Figure 3.

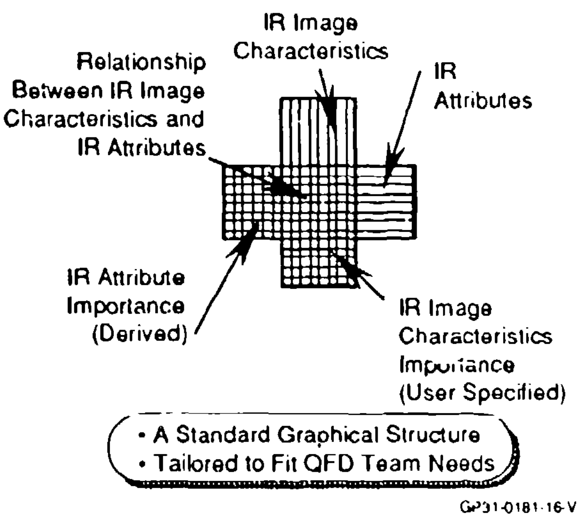


Figure 3. IR QFD Matrix Format

Input to the QFD matrix consisted of the Mean importance values for "Accurate Relative Intensities" and "Diurnal Effects" of IR backgrounds, and the "Infrared Signatures" of targets. Relationships between the IR image characteristics and the IR database attributes required to simulate them were defined as strong, medium, or weak.

The static IR database attributes were ranked in importance by the QFD evaluation function based on the user-specified IR image characteristic importances and the image-to-database attribute relationships[3,4]. The completed "Infrared Attribute Importance" QFD matrix is shown in Figure 4. It includes IR database attribute rankings for Initial Training, Advanced Training, and Mission Rehearsal. Note that the importance of the IR image attributes increases, as expected, when moving from Initial Training through Advanced Training to Mission Rehearsal. However, as described in the following sections, the relative ranking of the IR database attributes does not change significantly.

Reduced IR Attribute Set

The first step in the process of producing a reasonably sized set of IR attributes was to eliminate redundancies in the IR attribute superset derived from the IR models surveyed. This required the identification of attributes that were shared between IR simulation models, even if called by different names or using slightly different units. It is important to note here that "redundancy" is not the same as "dependency". Some dependencies among attributes were allowed to remain in order to simplify the usage of the SSDB. For example, although "heat capacity" is theoretically derivable from the other IR attributes and a geometric model, the task is not trivial. It is better to pre-compute the value and store it in the SSDB.

The second step was to separate the "static" feature and model attributes from "dynamic" (generally environmental) attributes. The static attributes include properties such as thermal mass or emissivity which generally remain constant for an object and are natural feature and model descriptors to be included in the SSDB. Dynamic attributes such as air temperature, cloud cover, humidity, wind speed, time-of-day, time-of-year, and other environmental attributes are important inputs to IR sensor simulations, but they are usually initialized or modified at simulation run-time.

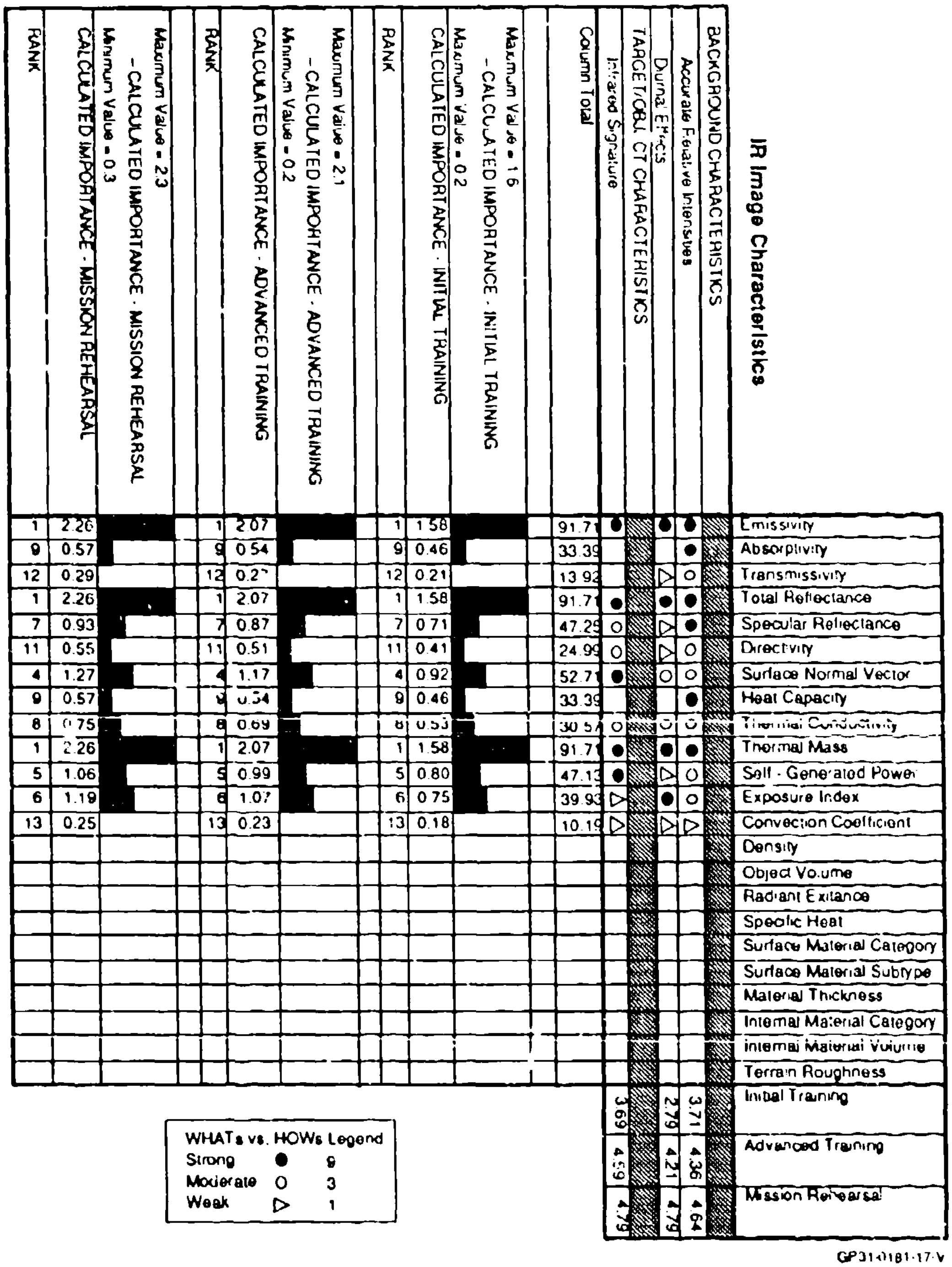


Figure 4. IR Attribute Importance QFD Matrix

The third step was to judiciously remove specific attributes that were extremely limited in scope, especially if they could be derived from the remaining set. The "derivability" criterion was not strictly enforced if other considerations prevailed. For example, although absorptivity and transmissivity have a complementary relationship, both are defined in the current SSDB and both are retained.

The reduced attribute list is:

Absorptivity
Convection Coefficient
Density
Directivity
Emissivity
Exposure Index
Heat Capacity
Internal Material Category
Internal Material Volume
Material Thickness
Object Volume
Radiant Exitance
Self-Generated Power
Specific Heat
Specular Reflectance
Surface Material Category
Surface Material Subtype
Surface Normal Vector
Terrain Roughness
Thermal Conductivity
Thermal Mass
Total Reflectance
Transmissivity

Emissivity, absorptivity, transmissivity, total reflectance, specular reflectance, and radiant exitance are wavelength dependent and should have separate values for visible (0.4 - 0.7 um), midwave IR (3.0 - 5.0 um), and longwave IR (8.0 - 12.0 um) wavebands.

Table 1 provides definitions and units for these attributes. The most common units used by the IR models have been retained wherever possible.

Prioritized IR Attribute Sets

QFD analysis was performed on the reduced attribute set using weighting factors corresponding to user inputs for the three training categories described earlier; Initial Training, Advanced Training, and Mission Rehearsal. The resulting ranked attribute lists are described below.

Initial Training - The ranked list of IR database attributes for Initial Training is:

1. Emissivity*
2. Total Reflectance*
3. Thermal Mass
4. Surface Normal Vector
5. Self-Generated Power
6. Exposure Index
7. Specular Reflectance*
8. Thermal Conductivity
9. Absorptivity*
10. Heat Capacity
11. Directivity
12. Transmissivity*
13. Convection Coefficient
14. Radiant Exitance*
15. Surface Material Category
16. Surface Material Subtype
17. Terrain Roughness
18. Specific Heat
19. Density
20. Object Volume
21. Material Thickness
22. Internal Material Category
23. Internal Material Volume

(* denotes wavelength dependence)

Attributes 14-23 are a reordered version of the attributes that are not given explicit rankings in the QFD matrix of Figure 4. This ordering emphasizes the remaining attributes that are most commonly used (e.g. Exitance) and/or are descriptive of backgrounds (e.g. Surface Material). The latter decision is based on the high importance of accurate IR background simulation consistently specified by the user community.

Advanced Training – The ranked list of IR database attributes for Advanced Training is the same as the previous list for Initial Training except that attributes 5 and 6, Self-Generated Power and Exposure Index, are reversed in priority. Although the absolute importance of all attributes increased with the overall importance of IR image accuracy, their relative importance remained essentially the same.

Mission Rehearsal – The ranked list of IR database attributes for Mission Rehearsal was identical to that for Advanced Training. Although the absolute importance of all attributes continued to increase with the overall importance of IR image accuracy, their relative importance remained the same. A closer look at the user-specified importance ratings of various image characteristics shows that Advanced Training and Mission Rehearsal are viewed as more similar to each other than either one is to the Initial Training Category. This is probably due to the heavy emphasis on the basic operation of IR sensor controls during Initial Training and the increased emphasis on image interpretation during Advanced Training and Mission Rehearsal.

Table 1. IR Attribute Definitions

Absorptivity	Ratio of energy absorbed to the energy incident.
Convection Coefficient	Convective heat transfer rate ($W/cm^2/deg\ C$).
Density	Mass per unit volume (gm/cm^3).
Directivity	Indicator of shape of the planar response curve of a feature or model for infrared. Three values: Omni-Directional, Bi-Directional, Uni-Directional.
Emissivity	Ratio of the rate of radiation from a feature as a consequence of its temperature only, to the corresponding rate of emission from a blackbody at the same temperature.
Exposure Index	The fraction of the surface of a material that is exposed to the sun, sky, wind and precipitation.
Heat Capacity	The amount of heat required to raise the temperature of a body by one degree, either at constant pressure or at constant volume and without inducing chemical changes or change of phase (cal/g/deg K).
Internal Material Category	Category code for Category material internal to an object (DMA codes).
Internal Material Volume	Amount of material inside an object (liters).
Material Thickness	Thickness of a material (cm).
Object Volume	Total internal volume of an object (liters).
Radiant Exitance	Rate of flow of radiation from an object per unit of surface area (W/cm^2).
Self Generated Power	Heat generated or removed from an object by other than radiation, conduction or convection (W/cm^2). Has positive value if object is emitting heat; negative value if object is absorbing heat (e.g. the cold side of a heat pump).
Specific Heat	The ratio of material's heat capacity to that of water.
Specular Reflectance	The ratio of incident to reflected energy normal to the surface.
Surface Material Category	Material code for the predominant material(s) making up the surface of a feature (DMA codes).
Surface Material Subtype	Subtype of Material Category.
Surface Normal Vector	The normalized vector perpendicular to a surface. The corresponding P2851 definitions are Vertex Normal and Polygon Normal.
Terrain Roughness	Roughness measurement for terrain.
Thermal Conductivity	Ability of a substance to conduct heat (W/cm/deg C).
Thermal Mass	A measure of the resistance of a material to changes in its thermal environment. A summary measure of the responsiveness of an object to heat (=Density * Heat Capacity * Material Thickness).
Total Reflectance	Ratio of total energy reflected by an object to the amount incident upon it.
Transmissivity	Ratio of energy transmitted by a feature or model to the amount of energy incident upon it.

GP31-0181-18-V

RECOMMENDED CONTENTS OF SSDB FOR IR SIMULATION

As a result of the QFD analysis and other user inputs, the following recommendations were made regarding the desired contents of the SSDB to support IR simulation. Since the attributes in GTDB and SIF are derived from SSDB, these recommendations also apply to GTDB and SIF.

Features and Models

All but two of the attributes in Table 1 should be in the SSDB attribute sets for features and models. Since the Terrain Roughness can be calculated from data already stored in the SSDB, it should not be stored separately. Also, since the Normal Vector can be calculated for existing polygons but may be variable for Constructive Solid Geometry (CSG)

quantities, its calculation should be deferred until GTDB or SIF files are produced, or left up to the user.

Internal Consistency

When radiation is incident upon an object, the Total Power Law states that[5]:

$$\text{Absorptivity} + \text{Total Reflectance} + \text{Transmissivity} = 1$$

If all three values are specified for any IR range, the total power law should be obeyed.

The Thermal Mass is a function of density, heat capacity, and surface thickness. If all four values are specified, the following equation should be obeyed:

$$\text{Thermal Mass} = \text{Density} * \text{Heat Capacity} * \text{Surface Thickness}$$

Environmental Attributes

Environmental attributes including cloud cover, air temperature, humidity, time-of-day, snow cover, haze/smoke/dust levels, and other dynamic attributes are important inputs to IR sensor simulations, but they are usually initialized or modified at simulation run-time, and are not recommended for inclusion in the core attribute set of the SSDB. Instead, the existing ability to define "Microdescriptors" describing weather and other environmental conditions can be used to store a set of initial values for these dynamic attributes, if so desired.

Inter-Feature Attributes

Some IR simulation models rely heavily on knowledge of the physical interfaces between adjacent features of an object to calculate heat flow across boundaries. Attributes required to support this approach include relationships such as inter-feature conduction, area of contact, and path length.

Such inter-feature relationships are not readily describable in the SSDB and deriving them from accurate geometric models, even when feasible, can be a formidable task. On the other hand, including such information explicitly in the SSDB for all related features of an object could demand a prohibitive amount of space. For this reason it is recommended that such inter-feature attributes be specified, when needed, in user-defined FACS fields.

SUMMARY AND CONCLUSION

In summary, in addition to the recommended set of 23 IR-related attributes, we have drawn the following conclusions from the user-survey and subsequent QFD analysis:

User Needs – The order of importance of IR simulation capabilities needed for most training applications are:

1) Sensor Controls Operation;
2) Accurate IR Backgrounds;
3) Accurate IR Target Signatures;
4) Other Sensor Effects.

Wavelength Dependence – Emissivity, absorptivity, transmissivity, total reflectance, specular reflectance, and radiant exitance are wavelength dependent and should have values stored for the visible (0.4 to 0.7 um), mid-wave IR (3.0 to 5.0 um), and long-wave IR (8.0 to 12.0 um) bands.

Environmental Attributes – Cloud cover, air temperature, humidity, and haze/smoke/dust levels are important inputs to IR sensor simulations, but are usually initialized at simulation run-time and are not recommended for inclusion in the core attribute set of the SSDB.

Inter-Feature Relationships – Some IR simulation models rely on knowledge of the physical interfaces between adjacent features of an object to calculate heat flow. Such inter-feature relationships are not readily describable in the SSDB, and could require a prohibitive amount of space.

ATR Applications – Although the ATR community represents a very important group of IR simulation users, it is unclear whether the SSDB can or should be expected to serve their very demanding needs. The requirements for extreme accuracy and completeness of IR predictions to test ATR algorithms and sensors under all conceivable conditions probably places the needs of the ATR community beyond the scope of P2851.

Availability of Attribute Values – Unfortunately, the difficulty of obtaining an IR attribute value appears to vary directly with its usefulness for predicting an IR image. In other words, the most useful "fundamental" attributes (e.g. emissivity, absorptivity)

are often the most difficult to obtain, while the more easily determined "superficial" attributes (e.g. radiant exitance, surface material category) are less robust. To support high-fidelity IR simulations over large gaming areas and/or to rapidly update P2851 IR simulation databases, it is essential that methods be developed to automatically extract values for these attributes from available data sources such as Multi-Spectral Imagery (MSI).

ACKNOWLEDGEMENTS

The authors would like to thank Gene Clayton, P2851 Program Manager at PRC, and his team for supporting this effort.

REFERENCES

1. Crane, P. M., "Forward-Looking Infrared Simulation Fidelity in Aircrew Training Devices", 13th Interservice / Industry Training Systems Conference, Orlando, Florida, 2-4 December 1991.

2. McDonnell Douglas Training Systems, "Infrared Simulation Database Attribution for Training", a study for DoD Project 2851, prepared for PRC, Inc. under Subcontract E01073-SC-92-004, 18 September 1992.

3. Akao, Y., "Quality Function Deployment - Integrating Customer Requirements into Product Design", Productivity Press, Cambridge, MA, 1990.

4. Sullivan, L.P., "Quality Function Deployment", Quality Process, June 1986.

5. Wolfe, W.L., and Zissis, G.J., "The Infrared Handbook", Infrared Information Analysis Center, ERIM, printed in cooperation with SPIE for the Office of Naval Research, Department of the Navy, Washington, D.C., 1989.

BEYOND VISUAL RANGE EXTENSIONS TO DIS

Ken Doris
Grace Mak-Cheng
Grumman Aerospace and Electronics

ABSTRACT

The successful 1992 I/ITSEC demonstration of DIS was a significant milestone in the development of the DIS protocols, proving that Version 1.0 of the standard is truly workable. Although the plans for the 1993 I/ITSEC demonstration focus on long-haul and live participant involvement, a vital ingredient to the eventual success of distributed simulation lies in the ability of subsequent versions of DIS to adequately support beyond visual range (BVR) encounters.

Simulation of BVR effects within the DIS context offers substantial increases to training effectiveness, tactics development, and improvements to the acquisition process. To achieve these goals we must overcome a new set of challenges. SIMNET, the predecessor to DIS, provided a solid background in the development of version 1.0 of DIS, but was limited to within visual range encounters. The BVR extensions found in DIS Version 2.0 can thus borrow little from the SIMNET legacy. New problems, such as sensor simulation, EW data base correlation, and environmental effects must be addressed.

This paper provides insight into the key issues associated with extending DIS to encompass the beyond visual range arena. In addition, it describes series of rapid prototype implementations of the Emitter, Transmitter, and Signal PDUs, starting with a joint Grumman/NTSC experiment held on the last day of the 1992 I/ITSEC show. The "lessons learned" from these implementations are discussed along with suggestions and guidelines for future development of BVR PDUs and associated data bases.

ABOUT THE AUTHORS

Kenneth Doris is a Technical Advisor in Grumman's Combat Systems organization. He is currently directing several research projects, including one devoted to DIS investigation. He is an active member of both the Communications Architecture and the Emissions subgroups of DIS. Last fall Mr. Doris lead the Grumman team at the 14^{th} I/ITSEC DIS demonstration held in San Antonio. He holds a Bachelor of Electrical Engineering from Rensselaer Polytechnic Institute and has twenty-five years experience in Simulation and C^3I, specializing in computer architecture and software engineering.

Grace Mak-Cheng is an Engineering Specialist in Grumman's Combat Systems organization. She is the Principal Investigator for an Independent Research and Development project devoted to DIS and networked simulation and actively participates in several DIS working groups . Ms. Mak-Cheng holds a Masters in Computer Science from New York Institute of Technology and a Bachelor of Electrical Engineering from New York Polytechnic Institute of Technology. She is an expert in the use of structured analysis and structured design software methodologies, especially as applied to embedded systems.

BEYOND VISUAL RANGE EXTENSIONS TO DIS

Ken Doris
Grace Mak-Cheng
Grumman Aerospace and Electronics

INTRODUCTION

DIS Version 1.0 (IEEE-93-1278) and its predecessor SIMNET have proven their worth in simulation of direct fire encounters, those in which the participants are within visual range of their opponents. Such encounters are the "end game" of any conflict and thus are vital to success. To reach that end game in a favorable position however, requires victory in conflicts which are beyond visual range (BVR). Consider the progress of Desert Storm. Its opening round was a series of attacks on Iraqi radar sites, denying the enemy many of its BVR "eyes". All Air Force and Navy strikes that followed were typically supported by EW jamming aircraft to further blind the enemy's electronic sensors. Iraq's main retaliatory tactic, the use of SCUD missiles, was thwarted primarily through the BVR success of the Patriot missile system.

Desert Storm cannot be taken as a model for all future conflicts since it was almost exclusively a land war. In the air and on/under the sea, Allied forces acted with virtual impunity - we cannot expect that luxury in the future. We must ensure that we are as well prepared to fight in those arenas as we were for the armored battles on land. SIMNET and DIS 1.0 have no provisions to support such conflicts. DIS Version 2.0 and beyond must provide these additional elements.

This paper explores some of the key issues associated with adding the BVR extensions to DIS. In addition, it describes series of rapid prototype implementations of the Emission, Transmitter, and Signal PDUs, starting with a joint Grumman/NTSC experiment held on the last day of the 1992 I/ITSEC show. The "lessons learned" from these implementations are discussed along with suggestions and guidelines for future development.

BEYOND VISUAL RANGE SIMULATION ISSUES

The DIS community is in the process of developing the additional message types (called Protocol Data Units or PDUs) and databases to support simulation of electronic, thermal, and acoustic emissions. Along with these there is a corresponding effort to develop techniques of simulating the environmental conditions that affect them.

Electronic Emissions Simulation

In the area of Electronic Emissions the DIS working groups have been concentrating on the simulation of radar and EW sensors and radio communications. In its current state the latest DIS standard defines three types of PDUs in this field:

- Emissions PDU
- Transmitter PDU
- Signal PDU

Emissions PDU

The Emission PDU has been under development since 1991[1]. At that time a Radar PDU also existed. As a result of the of the work performed by the Emissions subgroup of DIS, the Radar PDU was deleted and its requirements incorporated into the Emissions PDU (EMPDU). It was originally hoped that this single PDU could be designed to handle all types of emissions (including radar, EW, radios, infrared and acoustics). It quickly became evident that this would be unmanageable, and its scope was limited to radar, EW (near-term) and acoustics (future). Even given that limitation, the EMPDU is probably one of the most complex and little understood of all DIS PDUs.

The basic premise of the EMPDU is that every emitter is associated with a given entity. A *Simulation Host* is responsible for the simulation of the entity and its emitters (the entity may have one or more emitters "on-board"). The EMPDU

was therefore designed to represent multiple emitters for a single entity, and for each emitter, multiple beams, as shown in the simplified diagram below:

Emitting Entity ID
Number of Systems (N)
Emitter #1
Number of Beams (M_1)
Beam #1 Parametrics
:
Beam #M_1 Parametrics
Emitter #N
Number of Beams (M_2)
Beam #1 Parametrics
:
Beam #M_2 Parametrics

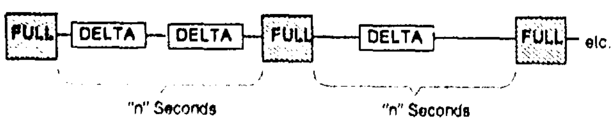
Simplified Emission PDU Diagram

That the PDU is of variable length is not by itself unusual for DIS messages, but in this particular case we find variable length records imbedded within other variable length records. This is not only difficult to comprehend at the design level, but also presents a formidable problem in software implementation.

During the early development of the EMPDU it was assumed that the entire PDU would be sent every time one or more parameters of any emitter and/or beam changed. It has recently been recognized that this would create a significant amount of traffic on the network (Emissions PDUs are among the largest in DIS).. A new set of rules was therefore established. Under these rules it is the responsibility of the host to:

- Issue a "Delta" form of the EMPDU whenever a change occurs to one or more of the emitters on the entity
- Issue a "Full" form of the PDU at a predetermined rate (e.g. every "n" seconds).

The Delta form of the PDU will contain data only for those emitter/beam combinations which have changed, while the Full form of the PDU will contain data for all emitter/beam combinations which are in the "on" state. Pictorially, the Emission PDU stream for a given entity might look like the following:



Typical Issuance of Emissions PDUs

Transmitter and Signal PDUs

These PDUs have been under development since 1991. At that time it was realized that the Emissions subgroup was overtaxed to handle this area and a separate subgroup was formed to independently attack this sphere of DIS simulation. The initial work of the Radio subgroup was based on earlier SIMNET experience and concentrated on the simulation of radios carried by land-based entities. Since that date the subgroup has developed this pair of PDUs (see simplified diagrams of the PDUs below) which seem to handle all of the requirements with the possible exception of data links (recently a separate subgroup of DIS has been formed to concentrate solely on the unique aspects of data links, with the possibility of either defining modifications to the existing PDUs or establishing new ones).

Emitting Entity ID
Radio ID
Radio Type
Transmit State
Antenna Location
Antenna Pattern
Transmitter Parametrics
Crypto System & Key ID
Number of Modulation Params (n)
Modulation Parameter #1
:
Modulation Parameter #n

Simplified Transmitter PDU Diagram

Emitting Entity ID
Radio ID
Encoding Scheme
Sample Rate
Length of Data (# of bits)
Number of Samples
Data Byte #1
:
Data Byte #n

Simplified Signal PDU Diagram

The concept of operation for the Transmitter and Signal PDUs is rather straightforward - each transmission from a radio is represented by at least two Transmitter PDUs (one for transmitter "on", the second for "off") acting as bookends around a series of Signal PDUs. Additional Transmit PDUs may be issued if any of the parameters of the transmitter change (e.g. antenna orientation) or after a fixed time interval

has expired. The number and size of the Signal PDUs vary depending on the type and length of data being sent.

In practice this concept is somewhat more difficult to implement, especially for voice. In transmitting voice the sending host must break the voice stream up into a series of individual Signal PDUs while the receiving host must reassemble them There is a trade-off that must be made between the number of packets, the processing burden placed on both the sender and receiver, speech intelligibility and transport delay. Theoretically the least burden would be to send one very large packet that contains all the speech. This would also result in excellent intelligibility, but would result in unacceptable delay at the receiving end. A compromise must be made in how the voice is packetized - maintaining reasonable delays and intelligibility along with moderate processing burdens. Experimentation with these factors is an ongoing effort in the DIS community.

Acoustic Sensor Simulation

This field is perhaps the most complex field to simulate and has no analogy in SIMNET or DIS 1.0. Acoustic are tightly linked to environment, especially in the ocean milieu. Unlike EM propagation that varies based on only a small subset of environmental conditions , underwater acoustic simulation must evaluate multiple conditions. A reasonably high fidelity acoustic model needs to consider the following environmental effects:

- Sea State
- Water temperature gradient (surface to bottom)
- Water salinity gradient (surface to bottom)
- bottom depth contour
- biological noise sources (snapping shrimp, dolphins, etc.)

In contrast a similar level of fidelity for Electromagnetic propagation needs only to look at rainfall rate and line of sight interference due to terrain. To draw an analogy with EM we would need to consider wind speed and direction at each given altitude, temperature gradients, as well as terrain surface roughness and time of day

The representation of acoustic emissions also requires simulation of such factors as the number of engines, rpm for each, cavitation of props (hull speed vs. prop rpm), and generation of knuckles as ships maneuver. The Emissions subgroup of DIS is only just starting to attack these problems.

Infrared Sensor Simulation

Desert Storm also highlighted one of our fortes in modern battle - our ability to "own the night". SIMNET and DIS Version 1.0 have only a rudimentary ability to show thermal imaging, based purely on simple rules to modify the normal daylight visual scenes produced by the image generators.. True simulation will require simulation of additional factors such as:

- Background thermal levels based on time of day and surface type
- Vehicle thermal level based on # engines, time on, time since off, color/surface type
- The effects of flares and IR jammers

Laser Designation Simulation

A key feature of our success in Desert Storm was the use of "smart weapons". Here again the ability to simulate these encounters in Simnet and DIS 1.0 is lacking. Our success in the Gulf War cannot be taken as predictive of the future.- The next aggressor may possess effective countermeasures. The key is to test our tactics against a more sophisticated adversary in this arena and to develop new tactics (and possibly weapon systems) to overcome such countermeasures

Data Base Issues

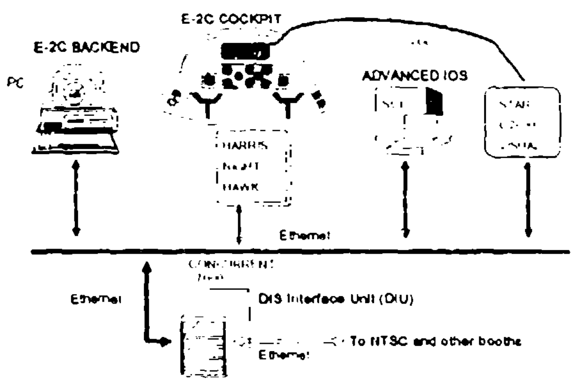
Environmental databases are being worked on by the individual (Air, Land, and Sea) subgroups of DIS, but are in very early stages.. Emitter databases are one of the tasks for the Emissions subgroup. Many exist, but all differ as they were developed for different purposes.. It is the charter of the Emissions Subgroup to define a standard set for DIS, and this work is ongoiing.

RAPID PROTYPING OF BVR PDUS

Over the past year Grumman has been involved in a number of rapid prototyping experiments with the Emissions and Radio PDUs, some via its own internal IRAD and others under contract to Navair/NTSC. The following paragraphs describe several of these activities.

Experimental test of Emitter PDU at 1992 I/ITSEC

From April of 1992 through August of 1992 a series of monthly meetings was held by the University of Central Florida's Institute for Simulation and Training (UCF/IST) to discuss the upcoming DIS demonstration at I/ITSEC. During the July meeting NTSC announced that it's portion of the demonstration would include an emulation of a SLQ-32 Radar Warning Receiver (RWR) and that they were interested in testing the use of the Emissions PDU to drive the RWR. Grumman decided to take advantage of the I/ITSEC opportunity and opened discussions with NTSC to jointly conduct a preliminary test of the PDU per the June 1992 format. A diagram of the Grumman I/ITSEC hardware configuration used during these tests is presented below.



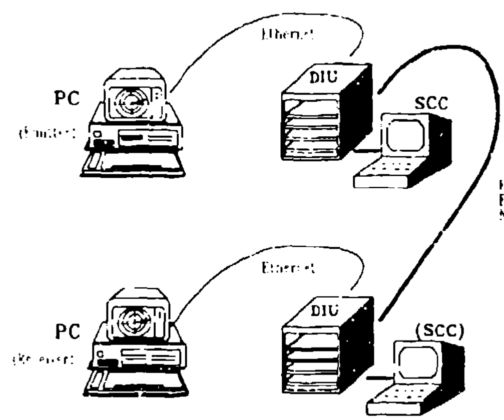
Grumman I/ITSEC DIS Demonstration Hardware Configuration

The format of the I/ITSEC DIS demonstration was rigidly set by UCF/IST and left only the last 3 hours on the last day (Wednesday, November 4) for what was termed "experimental PDU tests". It was during this time that Grumman and NTSC exercised the Emissions PDU for the first time.

The test consisted of Grumman's simulator acting as a hostile radar, issuing Emissions PDUs at a 1 Hz rate. To avoid security problems, the parameters selected for the radar were kept generic, and only the *Mode, Frequency and PRF* fields were evaluated by the NTSC simulation. The results of the test were successful, with the emulated SLQ-32 display correctly tracking the "hostile" E-2C as it moved around the gaming area. Unfortunately the selection of the last time slot at the show left too little time to extend the experiment to other variations of the Emission PDU (e.g. multiple emitters on the same platform, multiple beams per emitter, etc.). It is also interesting to note that the use of the Emission PDUs had no adverse effects on the other simulators on the network at that time.

Laboratory tests of Emitter, Transmit and Signal PDUs

Subsequent to the San Antonio demonstration Grumman extended its investigation of the Emissions PDU to include implementation and test of the September 1992 version of the Emissions PDU. This stage of the effort took place in Grumman's Great River, N.Y. facility, in a simplified version of the full testbed utilized at I/ITSEC. A block diagram of the Grumman DIS testbed hardware configuration is shown below.



Grumman's DIS Testbed

In this demonstration the two PCs act as two separate simulators linked over a DIS network. One PC acts as an entity carrying an emitter, while the other acts as an entity carrying a generic radar warning receiver. The case tested exercised one of the more complex cases supported by the Emissions PDU - that of radar emitters with a rotating beam. The beam rotates throughout a 360° arc, illuminating the receiving entity only once per rotation. The receiving system therefore must perform the following steps to correctly simulate the physics of the radiation:

1. Read the *Emitting Entity ID* field of the Emission PDU to determine which platform carries the emitter
2. Access that platform's latest Entity State PDU (stored locally) and read its *Position, Orientation* and *Time Stamp* fields to determine

the position and orientation of the platform at the time the Entity State PDU was issued

3. Dead Reckon the platform's position and orientation to correspond to the current time
4. Read the Emissions PDU's *Location with Respect to Entity*, *Beam Az Center, Beam Az Sweep, Beam El Center, Beam El Sweep* and *Beam Sweep Sync* fields to determine the position of the beam at the time the PDU was issued.
5. Dead Reckon the beam's position to correspond to current time.
6. Compare own ship position to determine if the receiver is being illuminated at the current time.

These steps must be performed at the local update rate of the receiver's simulation host computer and represent a fairly high computational load, giving further credence to the necessity of using a powerful DIU as the DIS interface at each host.

The results of the laboratory test of the September 1992 version of the Emissions PDU were successful. The receiving "simulator" correctly tracked the "radar beam" of the transmitting "simulator" as it moved throughout the gaming area.

Radio Voice PDU Implementation

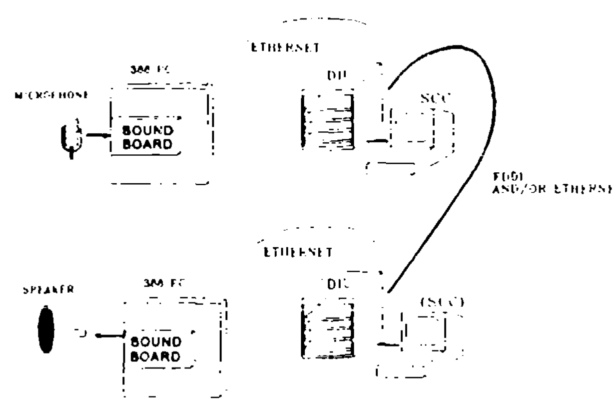
The simulation of voice radio traffic across computer networks is not new to DIS. Within SIMNET a SINCGARS radio simulation was implemented at Fort Knox, Kentucky and Fort Monmouth, New Jersey as early as 1989[2]. The SIMNET voice simulation differed from that required for DIS in one significant area - the method of voice encoding/decoding. It utilized a special board (developed by a SIMNET contractor) called a SIMVAD (SIMNET Voice Analog Digital), built specifically for that application. The board allowed for two types of encoding schemes, Adaptive Pulse-Code with Hybrid Quantization (APCHQ) or Continuously Variable Slope-Delta (CVSD). APCHQ is used for simulation of high quality speech at minimum bit rates, but incurs a high computational load. CVSD is lower quality and less compute-intensive and was found to be best at simulating the SINCGARS radio.

The method of voice encoding/decoding selected for DIS is called u-law Pulse Code Modulation (or u-law PCM). This technique was developed by Bell Laboratories and is used as the standard for digitizing voice on the Public Switched Telephone Network (PSTN) in all of North America. It is supported by a large number of commercial add-in boards for both PCs and workstations, and is of sufficiently high quality to support all voice applications.

For this investigation a goal was set to evaluate low cost commercial audio hardware to determine its suitability for DIS . After surveying a wide variety of commercial equipment, the hardware selected was the SoundBlaster, manufactured by Creative Labs Inc. This product is a PC compatible add-in board and is one of the most popular on the market. It supports a variety of sampling rates, and interfaces to standard microphones and speakers.

The major drawback to the board (and to all the low cost boards surveyed) is that was not designed for networked real-time operation as required by DIS. Thus our effort in this area concentrated on developing software that would allow the board to operate in a DIS environment.

A pair of boards was acquired and installed in the PCs in Grumman's DIS laboratory testbed, as illustrated below.



DIS Testbed Configuration for Voice PDU Experiments

The software developed provided for the following functions:

1. At the "simulator" originating the voice traffic:
   - PC Reading of microphone on/off switch, and issuing switch transition messages to the DIU across the local Ethernet connection.

- DIU Issuing of DIS Transmit PDUs upon microphone switch on/off transitions.
- PC Real-time memory buffering of incoming digitized voice originating from the SoundBlaster.
- PC packetizing of voice into Ethernet frames for transmission from PC to DIU.
- DIU unpacking of Ethernet frames and repacking into Signal PDU for transmission across the DIS network.

2. At the simulator receiving the voice traffic:
   - DIU reading of incoming Transmit PDUs (here additional logic would normally exist to determine if the local radio receiver was set to the correct communications channel, but was omitted within the limited scope of the experiment).
   - DIU unpacking of incoming Signal PDUs from the DIS network and repackaging for transmission to the PC across the local Ethernet connection.
   - PC unpacking of incoming Ethernet voice frames from the local DIU and buffering in memory.
   - PC outputs of voice memory buffers to SoundBlaster for subsequent output to loudspeaker.

The results of the experiment were a limited success in that voice was sent and received per the DIS standard, however the quality of the speech was poor and problems were encountered in processing all but extremely short messages. These shortcomings are disucssed in more detail in the "Lessons Learned" section.

Radio Data Link Implementation

For this implementation the following data links were considered as potential candidates for implementation:

- Link-11
- Link-4A
- Link-16 (JTIDS)

Of these Link-11 and Link-4A are carried by all E-2Cs and F-14s in service, while Link-16 is just becoming available on a subset of the aircraft thus far. Of Link-11 and Link-4A the latter was chosen for implementation since it was felt to be most representative of the interaction common to the two aircraft. An additional factor in its favor is that unlike Link-11, Link-4A is not encrypted (encryption is one of the controversial areas to be investigated by the new Data Link Subgroup).

The implementation of Link-4A utilized the same laboratory DIS testbed hardware configuration shown earlier. In this case the following software was developed to provide the functionality:

1. At the "simulator" originating the data link traffic:
   - PC reading of operator target hook commands and issuing track data messages to the DIU across the local Ethernet connection.
   - DIU unpacking of Ethernet frames at the DIU and repacking into Transmit and Signal PDUs for transmission across the DIS network.
2. At the simulator receiving the data link traffic:
   - DIU reading of incoming Transmit PDUs (here additional logic would normally exist to determine if the local Link-4A receiver was the destination specified, but was omitted within the limited scope of the experiment).
   - DIU unpacking of incoming Signal PDUs from the DIS network and repackaging for transmission to the PC across the local Ethernet connection.
   - PC unpacking of incoming Ethernet frames from the local DIU and buffering in memory.
   - PC reformatting of target position data and output to the simulated radar display.

The resulting demonstration confirmed that Link-4A target data can be transmitted across a DIS link using the current form of the Transmit and Signal PDUs. The format followed in the demonstration was as follows

1. At the transmitting "simulator" (PC) the radar operator hooks a target on the simulated radar display and selects the datalink message icon at the top of the screen, causing the "simulator" to send a Link-4A message to the receiving "simulator".
2. At the receiving "simulator" the radar scope display reflects the target the transmitting "simulator" has on its radar display. The operator can then hook that target and view its parameters.

LESSONS LEARNED

The implementation of the Emitter, Transmit, and Signal PDUs was not as straightforward as one might expect upon reading the DIS standard. For the Emitter PDU one of the major problems encountered was that ofimplementing the requirement to handle multiple targets within multiple beams within multiple emitters all in a single PDU. In our implementation for this project we went to the simplest solution by sizing our buffers to a reasonable maximum size. This is fine for entities with few (e.g. 5 or less) emitters, but future DIS exercises may require more elegant solutions (e.g. the use of multiple pointers to individual arrays in memory) to avoid tying up huge amounts of memory simply to handle a potential worst case.

For the Transmit PDU the main problem encountered was when one or more of these PDUs were lost due to simulator overload. Since these PDUs are sent in broadcast mode there is no guarantee of their arrival at each potential recipient. We found, for example, that during heavy voice traffic our PC "simulator" would sometimes lose PDUs. This is annoying if the PDU lost is a voice Signal PDU causing clipped speech, but is disastrous when a Transmit PDU was missed since it is used to determine whether to process the Signal PDUs. Entire portions of speech can be lost in this manner. In the lab our solution was to experiment with adding artificial delays between the DIU and the PC, a rudimentary form of flow control. The ideal solution is total flow control across the network, but that is apparently in conflict with the nature of DIS.

The voice version of the Signal PDU implementation was probably the most challenging. Our goal was to explore the potential of implementing DIS voice via mass produced audio hardware designed for the PCs market. The Soundblaster board we choose did not conform completely to the proposed DIS standard, nor was it designed with such an application in mind, but we felt its low cost and board support in the PC environment would offset that drawback. As stated earlier the results of the experiment are mixed. It proved that acceptable speech can be sent using the DIS PDUs with a reasonably short delay as long as the speaker keeps his messages very short (e.g. less than 2 seconds in length) and relatively infrequent. Long sentences caused the receiving PCs buffers to overflow, often causing an exit from the DIS session. We found a cure for this by introducing a delay at the receiving DIU-to-PC interface, however this caused a dramatic degradation of speech intelligibility. We believe that the use of more powerful PCs, such as 66 MHz 486s, would significantly improve the situation. The increase in computational speed, perhaps combined with *play-out protocols* to smooth the jitter between sucessive voice packets[3,4], would undoubtedly help, but we were unable to test these theories under the scope of this effort.

Another problem we observed affects both the voice and datalink implementations. The use of Ethernet 802.3 technology combined with the current format of the Signal PDU means that there is no guarantee that the received voice packets are in the same order that they were sent. This can be overcome by the use of a token-ring version of Ethernet or of FDDI, but these are considered too restrictive for DIS. A potential cure is to add a Signal PDU packet number to the existing set of fields and to locally reorder the packets based on this numbering scheme. Unfortunately the limited scope of our effort did not allow for experimentation with this concept to test its feasibility.

SUGGESTIONS FOR FUTURE BVR PDU AND DATA BASE DEVELOPMENT

One of the primary observations that can be made regarding the BVR PDUs is that they are fairly complex to implement. The current (2.03) version of the DIS standard provides only a top-level view of how to use them. Such PDUs require a separate "Users Guide" to aide the developers. The Emissions subgroup has started such a task but it will require a significant amount of work and time before it is sufficient for the DIS community.

Another observation is that these PDUs will become a major source of traffic on DIS networks. The use of delta vs. full PDU versions as described earlier may be required in all cases.

In the area of databases there is still much work to be accomplished. It appears that the establishment of "strawman" databases, at a declassified level, is the logical first step, but this is currently viewed as a low priority effort by the

DoD sponsors of DIS. The DoD customers must institute priorities in this area or it will continue to remain a background issue.

REFERENCES

1. Denver, J., "ECM/ECCM Implementation in the DIS Radar Environment", presented at the Fifth Workshop on Standards for the Interoperability of Defense Simulations, September 1991.

2. Gonzalez, J. "Electromagnetic Propagation Modeling for Distributed Simulation", Proceedings of the 13th Inter-service/Industry Training Systems Conference, November 1991.

3. Cohen, D., "Issues in Transnet Packetized Voice Communication", presented at the Fifth Symposium on Data Communications, 1977.

4. Gehl, T., "Packetized Voice for Simulated Command, Control, and Communications", Proceedings of the 13th Interservice/Industry Training Systems Conference, November 1991.

DEVELOPMENT AND APPLICATION OF THE EMISSIONS PROTOCOL STANDARD IN THE DIS NETWORK ENVIRONMENT

Jerry W. Denver
Boeing Defense & Space Group
Missiles and Space Division
Huntsville, Alabama

ABSTRACT

Ensuring interoperability of war-fighting simulations within the Distributed Interactive Simulation (DIS) environment requires network data protocol standards that ensure correlated effects in many diverse simulation arenas, not the least of which is the electronic interaction among detection, tracking, and jamming systems associated with the electromagnetic environment. Complicating the development of a general purpose emission protocol is the very broad nature of electromagnetic emissions and their inherent diversity between and within system applications. Other complicating issues include (1) interoperability of DIS with dissimilar equipment such as training devices, battlefield equipment, and war-gaming simulations, (2) extensive quantity of parametric data necessary to describe electromagnetic emissions, and (3) highly classified nature of the weapon system operational intelligence data.

An Emissions working group within the Workshop for Standards for the Interoperability of Defense Simulations is diligently forging ahead in its development of an approach that completely describes the electronic parameters in a radar emissions environment and ensures sufficient signal correlation among various types of simulation nodes. While various approaches have been considered, the approach taken is characterized with a strategy of broadcasting a minimum set of transmitting entity data, called the Emissions Protocol Data Unit (PDU), across a network medium. This data is coupled to a receiving node with a characteristic emissions data base containing static parameters that are specific to particular signal emissions. The protocol structure accommodates diversity in both radar emission types and various fidelity receiving node equipment, through the standard data structure organization as well as special purpose data fields, in order to accommodate unique equipment operational characteristics. Development of an Emissions PDU data structure is nearing completion with the expected submittal for standardization approval in late 1993.

This paper provides insight into the background of radar environment simulations in the DIS environment, discusses the interoperability issues relevant to various simulation approaches including data base functionality, and discusses the currently proposed Emissions PDU data structure and rationale for peculiar data fields. This paper also offers insight into the application of this PDU to various diverse radar system types and other Electronic Warfare (EW) emission types, and discusses areas where further investigations are warranted.

ABOUT THE AUTHOR

Jerry W. Denver is a principal systems engineer with the Simulation and Training Systems organization at the Boeing Defense & Space Group in Huntsville, Alabama. Currently, he is involved in research and development for advanced training systems. His previous responsibilities have included requirements definition and integration of Electronic Warfare (EW) software and database system simulations for Boeing's B-52 and B-1B Weapon System Trainer (WST) programs. He has been active in the DIS protocol standardization effort since 1991 through the Emissions subgroup. He authored the initial draft of the Emissions protocol for standardization considerations and has actively participated in the review/justification process including supporting development of subsequent modifications. Mr. Denver holds a Bachelor of Science degree in Electrical Engineering from Wichita State University.

DEVELOPMENT AND APPLICATION OF THE EMISSIONS PROTOCOL STANDARD IN THE DIS NETWORK ENVIRONMENT

Jerry W. Denver
Boeing Defense & Space Group
Missiles and Space Division
Huntsville, Alabama

INTRODUCTION

The Distributed Interactive Simulation (DIS) is a tri-service/industry sponsored initiative to develop standards for the development of interoperable defense simulations. DIS will allow the creation of simulated representations of totally interactive, war-gaming scenarios in a threat environment. The term threat environment applies to friendly and hostile weapon systems which can search, track, launch weapons, jam communication systems or otherwise act upon the affected EW systems. Thus, these threats operate and react according to tactics doctrine, including electronic counter-countermeasures (ECCM) tactics responses to applications of electronics countermeasures (ECM). Large quantities of threats in the scenarios will occur through the connection or "networking" of separate simulation components located both at local area networks (LAN) and at multiple, distributed locations (i.e. wide area network (WAN)). To provide interoperability of war-fighting simulations (both man-in-the-loop and computer simulations) within DIS, network data protocol and database standards must be developed to ensure correlated effects (electromagnetic, motion, and weapon) in many diverse simulation arenas. This includes realistic simulation of the electronic interaction among emission, detection, and jamming systems associated with the electromagnetic environment. Development of a general-purpose, electromagnetic interaction standard must effectively address the following characteristics associated with electromagnetic simulation environments:

a. Electronic Counter-Countermeasures

b. Electronic Emission Diversity

c. Data Parameter Quantity

d. Interoperability with Dissimilar/Variant Fidelity Devices

e. Classified Data

THE PROBLEM

Figure 1 illustrates the interactive "loop" of electromagnetic emissions, including both ECM and ECCM, which must be realistically simulated in order to account for in deterministic, real-time parameter changes associated with man-in-the-loop, "free play" simulation devices. In this one example of various possible emission "loops", the weapon system site's radar (hereafter called radar) illuminates the penetrator with an electromagnetic (EM) energy beam during a search/acquisition operational mode. With no ECM response from the penetrator, the radar transitions to a track mode. The penetrator platform's radar (hereafter called penetrator) then responds with specific ECM jamming signal characteristics which is effective against the radar and causes it to lose the penetrator as a tracked target (i.e. "breaklock"). The radar then uses its ECCM capabilities: electromagnetic signal characteristics that are inherently dynamic, either automatically or manually controlled by an operator, in a radar's operation. These variant characteristics (e.g. Radio Frequency (RF), Pulse Repetition Frequency (PRF), Pulse Width (PW)) represent data appropriate to specific electronic deception tactics used to attempt to defeat or negate the effectiveness of the applied ECM. In this example the radar uses a search/acquisition beam with different pulse characteristics, due to ECCM, which again transitions to a track mode when no response is received from the penetrator. The penetrator again applies the appropriate type of jamming ECM signal characteristics, which this time is not effective against the radar and is thus "defeated". So a "clear" (i.e. ineffectively jammed) environment is provided for the radar to track the penetrator and launch a missile or fire a projectile at it.

The inherent diversity of electromagnetic signals on the battlefield involve many unique attributes involving signal category, system type, system application, and system design. Signal category

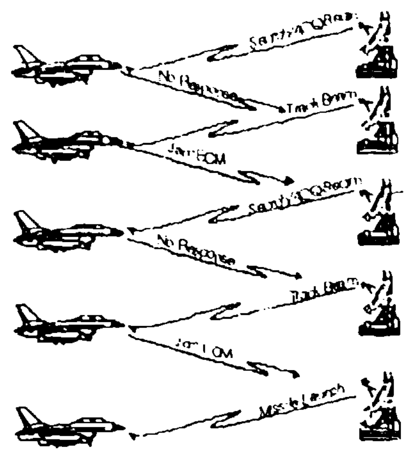


Figure 1 Electromagnetic Emission Interactivity

involves both voluntary or intentional (e.g. radar signal mainlobes) and involuntary or unintentional (e.g radar signal back scatter and infrared). System type addresses diversity inherent within radar, laser, infrared, sonar, etc. System application involves diverse purposes such as early warning, height finder, acquisition, tracking, and missile guidance. System design addresses diverse operational variances such as phased array antenna scans, multimode detection and tracking radar systems, and agility of pulsed parameters. The protocol standard must accommodate this ever-increasing diversity with a minimum of changes and, as a goal, with a minimum effort in the configuration management area.

The detailed description of electromagnetic emissions oftentimes requires an extensive quantity of parametric data, unique values for up to several hundred data types for each weapon system in a DIS environment, which is projected to include up to 10000 weapon system entities. The protocol standard must effectively address this complex situation, while minimizing the amount of real-time interface data transmitted across the WAN, due to the relatively slow speed and limited bandwidth of long haul network devices, including classified encryption devices.

Networking separate simulation components at distributed locations involves realistic performance through interoperability with both dissimilar and variant fidelity devices. These include actual military equipment (e.g. tanks, Non Line-of-Sight vehicles, test ranges, etc.), military simulation devices (e.g. aircraft, tanks, submarines, etc.), computer generated forces including war game models, and dismounted infantry. To ensure the widest possible application, the protocol standard must address the issues associated with interoperability of low cost, low fidelity simulation devices together with higher fidelity devices.

The simulation of a electromagnetic environment in a war-fighting scenario involves handling and processing of a wide range of weapon system data classifications: from unclassified to highly classified intelligence data. The highly classified portion pertains to the requirement for simulation scenarios that cannot be practiced in the "real world", due to the presence of hostile Electronic Intelligence (ELINT) signal data collection. Thus, the protocol standard must be able to support both classified and unclassified simulation nodes, and as a goal, be able to support classified operational scenarios on a totally unclassified network interconnection.

CANDIDATE SOLUTIONS

Viability of candidate solutions revolve around two key requirements:

1) Realistic support of ECCM tactics in an unscripted (i.e. freeplay) environment and

2) Minimization of real-time interface parameters to preclude excessive bandwidth loading upon the simulation network.

Figure 2 shows three candidate solutions investigated for supporting an emissions protocol standard. Approach A is the Resident Data Base Approach", whereby each simulation device on the network has emission parameter data bases whose number pointers (i.e. indices) are "common" to all receiving player nodes. This would allow a minimum set of parameter interface data, preferably a single pointer to a complete set of emissions data in the data base, to be "passed" to each simulation device. Approach B is the "Real-time Interface Data Approach", whereby each simulation device on the network transmits and receives all the necessary emissions characteristic data in real-time, thus negating the need for a resident data base. Approach C uses portions of Approaches A and B. It is the "Modified Real-time Interface Data Approach", whereby each simulation device on the network supplements the threat dynamic data (e.g. scan, ECCM parameters) it receives in real-time with a threat data base containing the remainder of the detailed, non-dynamic parameters.

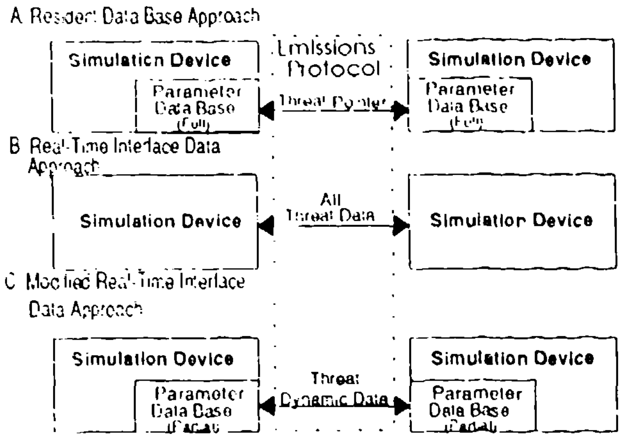


Figure 2 Candidate Solutions for the Emission Protocol Standard

Each of the three solutions offers significant advantages and disadvantages relative to the perceived requirements within the key categories (ECCM and data parameter quantity) as well as the other categories identified in The Problem. Figure 3 shows the performance requirement rating summaries for each approach according to the following criteria: green - meets/exceeds, yellow - marginal, and red - deficient.

	RATING CRITERIA				
Appr	ECCM (Key)	Data Param Qty (Key)	Diversity	Interop-erability	Unclass I/F Data (Goal)
A	Red	Green	Red	Green	Green
B	Green	Red	Green	Green	Red
C	Green	Yellow	Yellow	Green	Red

Figure 3 Candidate Solution Ratings

Approach A fully meets the requirements in three areas:

1) Unclassified operation across the long-haul network with classified local area nodes

2) Minimum amount of real-time interface data on the WAN

3) Interoperable with dissimilar devices.

However, it fails in one key area (ECCM) and also in diversity. In the key area, real-time changes in ECCM parameters (which support non-scripted scenario interactions between simulation nodes) cannot be effectively correlated between receiving nodes since all the parametric data is prestored in each node's characteristic data base. In the other area, support of diverse simulation elements is envisioned to be very restrictive in actual operation due to the need of compatible and correlated data bases at each simulation node, regardless of whether the node is low or high fidelity.

Approach B fully meets the requirements in three areas:

1) Accommodates real-time, unscripted ECCM weapon system operation

2) Accommodates emission diversity since resident data bases are not necessary

3) Interoperable with dissimilar devices.

However, it fails in one key area and conditionally fails another area. For the key area, the excessive amount of real-time interface data necessary to completely describe the electromagnetic signal parameters is envisioned to have a definite performance impact upon the LAN and most definitely the WAN, due to the relatively slow speed and limited bandwidth of network hardware devices. Due to it being a requirement "goal", it conditionally fails the support of classified operations across long haul networks, due to the need for classified encryption devices.

Approach C fully meets the requirements in two areas:

1) Accommodates real-time, unscripted ECCM weapon system operation

2) Interoperable with dissimilar devices.

However, it marginally meets the requirements in two areas, and conditionally fails another. It marginally meets the diversity requirement since the real-time dynamic data can be used, without the accompanying data base, to support a simulation node. This is envisioned to especially apply to lower fidelity simulation nodes. It marginally meets the key area of data parameter quantities, since there is envisioned to be a finite amount of dynamic data passed in real-time and supported by the remainder in a resident data base. Similar to Approach B, it also conditionally fails the support of classified operations across long haul networks.

In summary, Approach C - Modified Real-time Interface Data - was chosen because it either fully or marginally met all of the requirements except for unclassified interface data, which is a desirable but only a "goal" requirement.

PROTOCOL STRUCTURE DEVELOPMENT AND RATIONALE

With the candidate solution identified, development of the Emissions PDU detailed data structures proceeded, but was very difficult due to the myriad of perceived requirements, many times conflicting within themselves when taken as a whole. The following represents the overall requirement guidelines

High Level

1 Develop a general purpose protocol data structure that provides ample room for growth through inherent flexibility, i.e. applies to the emission types without changes in meaning or engineering units of associated parametric data field types.

2 Follow the DIS-accepted principle of "ground truth" (i e absolute) for describing the transmitting node's emitted energy.

3 Minimize the amount of interface data in the real-time protocol through

a) Maximum use of other protocol data structures already in existence, such as the Entity State PDU for associated platform motion characteristics and

b) Requirements for a resident emission parameter data base.

4 Ensure that the data structure doesn't inhibit or in any way restrict low fidelity simulations from easily participating. This includes providing enough data so that low fidelity simulations don't require access to supplementary data bases.

5 Identify a set of fundamental data (dynamic) for each EM beam of a transmitting system which will allow receivers to regenerate key EM parameters including the transmitting node's antenna scan volume as necessary.

6 Develop a "flat" data structure so the key data is quickly and easily accessible within the data structures. This will eliminate the "parsing" of data in multi-variant, layered field structures.

7. Ensure that the data structure efficiently supports the DIS parameter transmission philosophy of "send on change only" as well as "send all".

Lower Level

1. Provide "sorting" parameters for the receiving node's use in quickly determining EM emissions of interest.

2. Identify multiple weapon systems as-signed to a single geographic location, which is defined in the Entity State PDU.

3. Differentiate between multiple systems and beams of the same type at the same geographic location.

4. Correlate multiple EM beams, as applicable, to a single weapon system operating in each of its operational modes. This is necessary for accuracy and efficiency in emission correlation at the receiving nodes, since pulse-to-pulse emission timing descriptions are not provided.

5. Identify the variant individual location of multiple emitting systems tied to an entity (e.g. a ship or an aircraft platform) described at a single geographic location.

6. For a "slow scanning" radar system, identify the real-time location of the transmitting node's main lobe EM beam within the antenna system scan volume.

7. Provide pointers/indices to detailed invariant parameter sets within the emissions characteristics data base.

With these requirements in hand (which have evolved over the last two years), the Emissions protocol data structure has been developed and continually refined throughout that same time period.

Figure 4 identifies the current data structure, achieved through consensus in DIS/Emissions subgroup meetings.

Emission PDU Fields	
PDU Header	Protocol version - 8 bit enumeration Exercise ID - 8 bit unsigned integer PDU Type- 8 bit enumeration Padding - 8 bit unused Time Stamp - 32 bit unsigned integer Length - 16 bit unsigned integer Padding - 16 bit unused
Emitting Entity ID	Site - 16 bit unsigned integer Application - 16 bit unsigned integer Entity - 16 bit unsigned integer
Event ID	Site - 16 bit unsigned integer Application - 16 bit unsigned integer Entity - 16 bit unsigned integer
State Update Indicator	8-bit enumeration
Number of Systems (N)	8-bit unsigned integer
Padding	16 bits unused
System Data Length	8-bit unsigned integer
Number of Beams (M)	8-bit unsigned integer
Padding	16 bits unused
Emitter System	Emitter Name-16-bit enumeration Function - 8 bit unsigned integer Emitter ID - 8 bit unsigned integer
Location (with respect to entity)	x - 32 bit floating point y - 32 bit floating point z - 32 bit floating point
Beam Data Length	8-bit unsigned integer
Beam ID Number	8-bit unsigned integer
Beam Parameter Index	16-bit unsigned integer
Fundamental Parameter Data	Frequency - 32 bit floating point Frequency Range - 32 bit floating point ERP - 32 bit floating point PRF - 32 bit floating point Beam AZ Center - 32 bit floating point Beam AZ Sweep - 32 bit floating point Beam EL Center - 32 bit floating point Beam EL Sweep - 32 bit floating point Beam Sweep SYNC - 32-bit floating point
Beam Function	8 - bit unsigned integer
Number of Targets in Track/Jam Field (P)	8 - bit unsigned integer
High Density Track/Jam	8 - bit unsigned integer
Padding	8 - bit unsigned integer
Jam Mode Sequence	32-bit unsigned integer
Track/Jam	Site - 16 bit unsigned integer Application - 16 bit unsigned integer Entity - 16 - bit unsigned integer Emitter ID - 8 bit unsigned integer Beam ID - 8-bit unsigned integer

Figure 4 Emission PDU Structure

However, compromise to requirement guidelines, as shown in Figure 5, have been necessary in the following two areas:

Variant Length Data Structure

The Emissions protocol data structure is set up as a variant length data structure having the capability or the following major characteristics: a single or a multiple number of emitting systems (N) "tied" to a specific geographic location, each system having the capability for multiple EM beams (M), simultaneously operating with unique dynamic parameters (called the Fundamental Parameter Data), and each EM beam designating targets (P) which are being "tracked" or "jammed". Thus, three layers of variant data length are accommodated in the data structure.

Ground Truth

The Emissions protocol data structure deviates from the "ground truth" principle by identifying

Requirement	Emission PDU Field Structure Design
1. Multiple Weapon System / Single Location	1. a. Number of Systems(N) b. Emitter System
2. Variant Individual Locations	2. Location (with respect to entity)
3. Multiple System/Beam Differentiation	3. a. Emitter System- Emitter ID Number b. Beam ID Number
4. Sorting Parameters	4. a. Emitter System b. Beam Function
5. Fundamental Dynamic Data	5. Fundamental Parameter Data
6. Real-Time Antenna Beam Location	6. Fundamental Parameter Data - Beam Sweep Sync
7. Identify Pointers to Parameter Sets	7. a. Beam Parameter Index b. Jamming Mode Sequence
8. Track/Jam Data	8. a. Number of Targets in Track/Jam Field (P) b. High Density Track/Jam c. Track/Jam Identify
9. Low Fidelity/Supplementary Data Base Not Needed	9. Fundamental Parameter Data - Frequency Range - Effective Radiated Power (ERP)
10. Send on Change Only	10. a. Number of Systems (N) b. Number of Beams (M)

Figure 5 Emission PDU Requirement Versus Design Traceability

targets being tracked or jammed to ensure a "fair fight" on the battlefield. This will differentiate platforms which would be illuminated by a large amount of energy, such as in track while scan modes of phased array radars and multi-target jammers. This is intended to preclude signal correlation problems, which may occur due to "time delays in the network" and lack of computational resources in lower fidelity simulations. To minimize the potential for extended and invariant data lengths associated with tracking and jamming target lists, a number of control fields have been inserted. These are the number of targets within the sweep volume, the identification of the targets, and the high density track/jam indicator field. This latter field is used in cases where the number of targets being illuminated exceeds a constant (currently envisioned to be five), thereby informing all receiving targets within the beam sweep volume that they should assume tracking or jamming by the associated transmitting node.

DATABASE STRUCTURE DEVELOPMENT AND RATIONALE

As stated previously, a complete description of the EM environment for each player in the simulation exercise requires the receipt of the Emissions PDU across the network and an accompanying emissions data base resident at the receiving simulation. To ensure that each simulation player has the same or a subset of the same identical emission parameters for all simulation players in the simulation exercise, an off-line (i.e. non-real time) data base generation process is envisioned as shown in Figure 6.

The process starts with source data containing known, validated emission parameter data base(s). This data source could be one or more of the many data bases available from the Navy or Air Force. However, the DIS/Emissions subgroup currently thinks the Electronic Warfare Integrated Reprogramming (EWIR) database, generated by the Defense Intelligence Agency (DIA) representing the DoD official position for complete and validated emissions data, is the most probable candidate for a single source of data.

Then, a DIS data base needs to be created through data extraction and transformation of the source data. This is necessary because EWIR data textual formats are not compatible with simulation data formats, especially in the area of parameter links and nomenclature corresponding to unique beam emissions. Thus, unique beam identification for DIS application is envisioned. The DIS data base should also be able to reflect data inputs from the "user", in order to respond efficiently to unvalidated data in "what if" exercises.

From this DIS data base which is all-inclusive of all simulation parameters necessary to support any simulation node, another process is envisioned whereby data is extracted from it to produce unique simulation data bases for each simulation player or entity. Each simulation data base could be built according to the unique requirements of each simulation, thus the size of the data bases could be different due to different systems and parameters being included in the data base. However, each coincident parameter would be the same in each data base as data reformatting during the extraction process is not foreseen as a necessary or desirable feature. Since this data extraction process also effectively produces the parameter indices, correlation of data parameter indices as referenced by the Emissions PDU is ensured among simulation nodes.

Definition of the DIS data base structure and transformations necessary to produce it is currently in the conceptual stage. Its development is proceeding in parallel, through an outside contracting agency, with the development of the Emissions PDU through the DIS/Emissions Subgroup.

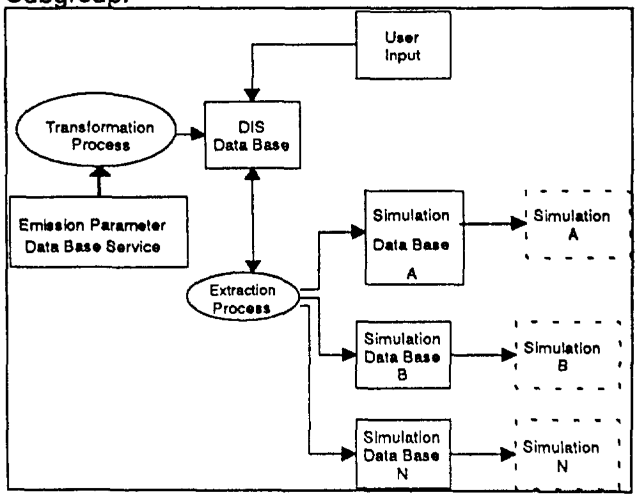


Figure 6 Emission Data Base Generation

APPLICATIONS AND CONCLUSIONS

The Emissions protocol standard is intended to apply to a broad array of electromagnetic emissions (See Figure 7) within the system application

categories of radar, jammer, and navigation systems. However, the emphasis to date has been to develop applicability to both standard and exotic system types within the general categories of radars and jammers.

Emission Type	Emission Mode	Protocol Definition
• Radar		
-RF Fire Control & Missile RF Seeker	Pulsed, CW (Incl ESAR)	Emission
- Missile (Beacon, Fuze)	Pulsed, CW	TBD
- RF Bistatic	Pulsed, CW	Emission
- Laser	Pulsed	TBD
• Jammer		
-RF	Noise, Deception	Emission
-IR	Pulsed Noise	Emission
- Laser	TBD	TBD
- Chaff	N/A	In Process
- Flare	N/A	In Process
• Navigation		
- Radar (TF, Ground Map)	Pulsed	Emission
- IFF	Pulsed	In Process
- TACAN	Pulsed	TBD
- ILS	Pulsed	TBD
- Ground Beacon	Pulsed, CW	TBD

Figure 7 Emission Types & Modes vs. Protocol Definition

Radar

Within the radar category, there is inherent broad diversification of functionality, usage, and characteristics. Notwithstanding, the protocol standard is envisioned to address both continuous wave (CW) and pulsed RF fire control systems including any associated missile RF seekers. Further, it addresses both mechanically and electronically steered phased array radar (ESAR) antenna systems directly through the protocol. Nuances in parameter agility (e.g. pulse, frequency, scan, etc.) inherent in EM beams are addressed in the accompanying emissions data base.

The protocol standard is intended to apply to unique emission capabilities such as bi-static radar. A possible approach is, for each target within the transmitting simulation node's field of regard, to define a radar beam as having a bi-static function with signal parameters as defined in the protocol with the exception of effective radiated power (ERP), which is normally defined at the transmitting antenna's output. In this case, the ERP is defined at the target, thus including the effects of one-way atmospheric propagation loss. Then each passive receiver simulation node, upon finding bi-static transmitting nodes with targets, can model the EM energy radiated in their direction based upon their perception of target radar cross-section and one way propagation loss. The transmitting node defines a second beam, as necessary, to be the reference signal for the passive receiver.

Application of the Emissions protocol to other radar system types such as laser radar and missile associated signals (e.g. beacon and fuze), while thought to be directly applicable, is still being investigated.

Jammer

Within the jammer category, the protocol standard is envisioned to address both RF and infrared (IR) systems; operating in standard noise (both wide band and pulsed) modes as applicable to the peculiar system. More exotic jamming operational sequences such as pulsed deception repeater modes are also supported through the use of the "Jamming Mode Sequence" field, which is a reference pointer to detailed jamming emission parameter characteristics in the resident data base.

Application of the Emissions protocol to other jammer system types such as laser jammer and expendables (e.g. chaff and flare) is still being investigated. Application to expendables, especially chaff, doesn't appear likely as separate data protocol structures appear warranted.

Navigation

Within the navigation category, the protocol standard is envisioned to address RF systems; operating in standard navigation modes (e.g. ground map, terrain following) as applicable. Exotic ECCM nuances in parameter agility (e.g. pulse, frequency, scan, etc.) inherent in operational EM beams can be accessed in the detailed navigation emission parameter characteristics in the resident data base.

Application of the Emissions protocol to other navigation system types such as Tactical Air Navigation (TACAN), Instrument Landing System (ILS), and Ground Beacons is still to be investigated. Application to International Friend or Foe (IFF) is currently being investigated.

SUMMARY

This paper describes the continuing development and application of the Emissions Protocol Data Unit (PDU) data structure standard in the DIS network environment. This protocol, when combined with emission data bases at receiving entity nodes, incorporates signal detection and jamming capabilities into an interactive ECM and ECCM environment through various means. First, it provides interface data for dynamic site operational characteristics which can be correlated across all network entities in both "clear" and "ECM/ECCM" environments. Second, it provides ECCM tactic "pointers" for which detailed data is contained in the emissions parameter data base. Third, through the "Tracked/Jammed ID" field, it provides a means to identify targets within radar track and jamming field coverage, to better ensure "fair fight" correlation and support low fidelity simulation nodes. Lastly, through the system/beam data structure, it supports the existence and correlation of transmitting systems operating in transmission modes with multiple radiation beams of energy.

While the Emissions protocol structure presented here represents a good start in addressing the multi-variant needs of the emissions area, there needs to be continued investigations in network implementation issues as well as key interactive features and other portions of the electromagnetic interaction environment. The network implementation issues involve:

1) Timely transmission of classified emissions data across a WAN,

2) Size of the protocol structure data packets which can be transmitted on the DIS network, and

3) Performance impacts of bandwidth loading due to the size, frequency, and weapon system player quantities envisioned on the network.

A key simulation feature supporting realistic interactive environments that needs to be addressed is the correlation of environmental effects (e.g. object occulting, atmosphere) with the EM emissions processed by the receiving entity nodes. Other portions of the EM environment need further investigation before a decision of the application of the Emissions protocol can be made. These include involuntary emissions (e.g. infrared signatures, acoustics) as well as portions of the voluntary emissions. The voluntary emissions identified to date as the next likely candidates for application in the DIS emissions environment through expansion of the currently proposed protocol or development of new emissions protocols include peculiar navigational radar (e.g. IFF), jamming expendables (e.g. chaff and flare), and sonar.

ACKNOWLEDGMENT

The author wishes to express his thanks to all the members of the Emissions subgroup of the Integrated/Time Mission Critical group for their efforts in the review and continued development of the Emissions Protocol for the Distributed Interactive Simulation (DIS) standards initiative.

REFERENCES

(1) Standard for Information Technology - Protocols for Distributed Interactive Simulation Applications, Version 2.0 Second Draft, March 22, 1993

HAVING EQUAL SIMULATORS DOES NOT GUARANTEE A FAIR FIGHT

Roy Latham
Computer Graphics Systems Development Corporation
Mt. View, California 94043-2330

ABSTRACT

Having a *fair fight* in networked simulation means the outcome is determined by the quality of the user's skill, tactics, and modeled-real-world equipment, and not by the limitations and peculiarities of the user's simulation equipment. It might seem that by ensuring that all participants in an exercise have equal, or identical, simulators one would 'level the playing field' with respect to any shortcomings in simulator realism, but it turns out that this is not true. A way to treat these problems is to analyze the role of each player in a simulation exercise against the objectives of the exercise. This analysis is aided by charts that help relate image generator characteristics to the requirements of the simulation exercise. Even a simplified analysis may produce simulation results having greater validity than obtained by attempts to equalize simulator equipment. Future efforts at verification and validation will be aided by developing catalogs of simulator capabilities, standard task descriptions and their simulation requirements, and templated methods of analysis.

ABOUT THE AUTHOR

Roy Latham is president of Computer Graphics Systems Development Corp., a consulting firm specializing in real time graphics and simulation systems technology. Previously, he managed graphics projects at Sun Microsystems, Kaiser Electronics, and the Singer-Link Flight Simulation Division. He holds patents in simulator visual systems, has published numerous papers in the field of training and simulation, and has authored the book, *The Dictionary of Computer Graphics Technology and Applications*. Mr. Latham holds BS degrees in Electrical Engineering and in Aeronautics and Astronautics from M.I.T., an MS in Applied Mathematics from the State University of New York at Stony Brook, and an MS in Computer Science from the University of Santa Clara. He is a registered professional engineer and licensed patent agent.

HAVING EQUAL SIMULATORS DOES NOT GUARANTEE A FAIR FIGHT

Roy Latham
Computer Graphics Systems Development Corporation
Mt. View, California 94043-2330

"FAIR FIGHTS" MAY BE UNEQUAL

Networked simulation, using a combination of manned simulators and computer generated forces, is being increasingly used for critical evaluation of weapons systems and tactics. For such evaluations, new elements are inserted into a simulated battlefield situation to critically observe the outcome. The fair fight problem is that of ensuring that the outcome is primarily determined by the characteristics of the systems and tactics under evaluation, and not by limitations or artifacts in the simulation systems.

The phrase "fair fight" is quite misleading in this context. The objective is not that every participant have an equal opportunity to win, but rather that whatever the chances would be in the real world those chances should be accurately reflected in the simulation. The terminology is well established, and quite probably misleads non-specialists into jumping to the conclusions as to what the problem is and how it ought to be solved. It is more appropriate to refer to the "network validation" problem rather than the "fair fight" problem.

Another misconception is that when unequal simulators are matched, the advantage will likely go to the more expensive higher-fidelity simulator. Higher fidelity is likely to mean higher visual resolution, wider fields of view, and faster update rates, which are indeed advantages. However, higher fidelity also is likely to mean much better database cover and concealment, textured surfaces that lower target contrast, and more effective obscuration by weather, dust, and smoke. One study has been cited as showing that a lower fidelity simulator achieved an advantage simply by relaxed criteria for scoring hits. [1] Overall, it will depend on the details of the circumstances whether the higher fidelity or lower fidelity simulator has an advantage, all other things being equal.

USING EQUAL SIMULATORS

An apparent, but incorrect, solution to the fair fight problem is to provide everyone in the simulation with the same, or equivalent, equipment. There are many reasons why the concept of applying equal simulators does not solve the fair fight problem. Briefly, here are three reasons.

1. Visual systems use level-of-detail switching techniques to provide more detail near the eyepoint and less detail in the distance. Because the scene is position dependent, even identical simulators do not provide equal lines of sight
2. All simulators have limited fidelity, and how the limitations affect the outcome of an encounter depends on how those limitations affect one set of tactics and systems versus how those same limitations affect a different set of tactics and systems.
3. Different roles in a simulation inherently require substantially different levels of fidelity. This point may be subtle if the exercise only involves, say, armor units, but it should be obvious that fair fight considerations do not require giving the same visual system to an attack helicopter and a submarine involved in the same exercise.

The first of these points has been discussed previously [2]. The latter points deserve some elaboration.

Suppose we wish to evaluate the combat effectiveness of a new sensor's improved ability to penetrate smoke and haze. Now, suppose that all the simulators used in the evaluation have a severely limited ability to depict smoke and haze. Even though all the simulators are equal in this regard, it will not be possible to do a fair evaluation of the new system.

To make a fair evaluation of our new smoke-penetrating sensor we must focus more closely on the objectives of the evaluation. One objective might be to see if ordinary cover and concealment from landscape features under certain scenarios actually made the smoke penetrating capability of limited use under the relatively long acquisition ranges of interest. Another objective might be to evaluate the best mix of the new weapon with existing weapons, and to see how it could be used most effectively. More limited objectives might be to study human factors issues related to operating the weapon, such as how long it takes to set up and operate in various conditions.

None of these objectives require that other units in the networked simulation have improved abilities to depict smoke and haze. Other objectives could very well require that other players have improved capabilities in this regard; we might, for example, want to evaluate a future battlefield scenario in which all forces had upgraded sensors. But with specific objectives as stated, it is apparent that there is no need to upgrade the capabilities of simulators in the exercise which do not have sensors that require those capabilities. For the limited objectives of the exercise, the other units in the exercise are mostly serving in the role of targets for the new system. The role of target may not be an interesting one, but it is a valid requirement for a simulation with limited objectives, and the role can clearly be played effectively with simulators having a much different level of fidelity than that of the system under evaluation.

The results of a simulation performed against these limited objectives in the scenario described would have to be suitably limited. But the results of any simulation need to be analyzed with the limitations of the simulators involved. For example, whether the simulators were equal or unequal, we would always have to ask if the fidelity of the simulation was high enough to provide realistic cover and concealment to play against the long range characteristics of the weapon.

The example cited above is not entirely hypothetical. A similar simulation evaluation was recently proposed. A simulator with higher sensor fidelity was proposed to be mixed with simulators of lower sensor fidelity. The mixed fidelity simulation was in fact rejected by government reviewers on "fair fight" grounds, with the point made that all other simulators involved would have to be upgraded to make the experiment valid. The rejection stuck.

Even though the concept of "equal simulators make a fair fight" is wrong, the simplicity of the idea is a strong attraction. It is, in a way, a tribute to the recent successes of networked simulation that important decisions are being made at levels high enough to be out of the hands of specialists. However, there is a distinct possibility that simulation professionals will not be able to overcome common misperceptions, at least not in the near term.

Sticking to the wrong concept unnecessarily limits the use of hundreds of millions of dollars in simulation assets. SIMNET equipment was not necessarily designed with systems-evaluation tasks in mind. The limited fidelity of SIMNET equipment can nonetheless be used judiciously in a mix with higher fidelity simulations to produce valid experiments achieving specific objectives. Instead, there is a strong push to have degraded modes in higher fidelity simulators, again citing fair fight concerns.

Since there are many objectives that can only be met with higher fidelity simulators, simulation experiments that could be effectively run in a mixed-fidelity mode will have to be avoided. Put to the test, no one really believes that equal simulators guarantee a fair fight. For example, no one would suppose that SIMNET equipment in its present form could be used for effective evaluation of the state-of-the-art sensors in combat scenarios, even though all the simulators are equal.

Equal Sims Useful for Competitions

There are limited circumstances when the notion of having equal simulators is applicable. If individuals or groups are being graded for their performance using simulator equipment in the performance of the same tasks under the same circumstances, the equipment will sometimes have to be closely matched. For example, suppose two groups of soldiers are given classroom training according to different curricula. The objective is then to test, using networked simulation, which group was better able to translate its classroom training into practice. In such a case, having simulators of differing capabilities would be a distracting complication. It would probably be acceptable to use a mix of equipment if each group had the same mix, with the different types of simulators assigned to corresponding roles in the two groups.

Even in the case where individuals are being graded on their performance through use of a simulator, the case for equal simulators is not always conclusive. For example, tests for drivers licenses and pilots licenses are given using a variety of different real vehicles in each case. Everyone assumes that the fundamental skills being tested are independent of the vehicle details, and that the test evaluator can grade performance in a variety of circumstances.

Because simulators are traditionally associated with training, and training is traditionally associated with testing or competition, there may be the erroneous belief that these circumstances are typical. Such circumstances are not common, because even in the use of simulators to teach team tactics, the participants are performing different roles in the overall scenario, so the judgment of an instructor will be required to take into account the differences of assignment. To demand equal equipment, the situation has to be along the lines

of a computer game with machine scoring. Our discussion mainly concerns the use of simulators for evaluation of systems and tactics, a much different situation.

USING MISSION ANALYSIS

The correct approach to building a valid simulation network exercise is to analyze the simulation system requirements against the objectives of the simulation. The difficulty of performing such an analysis goes some ways towards explaining why the invalid, but much simpler, approach of blindly matching simulator characteristics remains such a potent alternative. The best we can do is to develop tools and procedures that simplify the analysis task and make using the correct approach less onerous.

One of the strongest aspects of the military and aerospace approach to problem solving is its focus on objectives, and on the need to derive requirements from objectives. A requirements-oriented approach is exactly what is needed to derive simulation requirements. The obstacle is that performing the analysis may be tedious and difficult. However, there is no escaping the need to relate simulation objectives to simulator requirements, and even a simplified analysis of simulation objectives will yield far better results than blindly requiring equal simulators.

The analysis can be broken into steps:

- List the objectives of the simulation work
- Derive the exercise scenarios that must be played out to generate the data necessary to achieve each of the objectives
- Define the players and roles required to play out the exercise scenarios
- Define the tasks to be performed by each player in the required roles
- Establish the simulator requirements to perform the roles
- Match the simulator requirements with the inventory of available equipment
- Specify new hardware and software to meet simulation requirements that cannot be met with available resources, or limit the objectives to be consistent with the resource

Terminology

In the forgoing, an *exercise* is a session in which networked simulators interact on line. An *objective* is a purpose of the exercise; the objective is often to answer a question about the effectiveness of a new system or procedure. A *player* is a participant in the simulation, either a manned simulator representing a vehicle, a manned control element, another type of simulator, or, nowadays, a comput- controlled version of a simulated unit.

The *role* e player is what the player is expected to do , the context of the simulator exercise. The role in a particular simulation may be far less than its full capabilities, if a simulation exercise is focusing on logistics, for example, a complex weapons system may have no role in that particular exercise other than to receive and consume fuel. A *task* is a skill or capability required to perform a role. For example, if a role involves moving a vehicle to a particular location, certain types of navigation skills and capabilities will be required to accomplish the role. Finally, a simulator *requirement* is the technical specification which a simulator or simulation system must meet to permit the player to accomplish a set of tasks for a required role in the simulation. If the task is navigating by landmarks, the simulator must provide sufficient visual cues to accomplish that task, and this will ultimately be reflected in a simulator visual system requirement to produce a certain number of polygons with a certain amount of terrain detail and a certain number of man-made features.

Analysis Methods

This type of analysis is commonplace for training system procurements. There are many fine examples of such analyses yielding effective systems and experiments. The best efforts are the products of collaborations among users (who help set objectives and establish roles), human factors specialists (who help relate roles to tasks and tasks to requirements), and simulation technical specialists (who help relate requirements to equipment capabilities, and to design new capabilities).

It is more feasible to establish a team and perform the analysis for a large training system procurement than it is for a smaller scale systems research project. A major procurement will have a large financial incentive not to overspecify or underspecify the system. Training objectives do evolve over time, but the initial analysis can be constrained to a particular training mission that has been established with care.

Exactly the same type of analysis that relates objectives to simulator requirements for training should be applied to each use of networked simulation for research or systems development. For such widespread use the analysis process needs to be made as simple and efficient as possible. Otherwise, erroneous, oversimplified methods may predominate. In particular, if the simulation

community does not work energetically to get rid of the erroneous "equal simulators" concept, we will spend money needlessly upgrading simulators, and nonetheless conduct invalid experiments. Invalid experiments will most likely mislead the development of new weapons systems and doctrine.

Role and Simulator Cataloging

Much of the time required for a full analysis could be saved by cataloging frequently used data and providing tools for retrieving and combining the data.

From the analysis steps, observe that roles translate to requirements. Consequently, it makes sense to save the results of role analyses and put them into a database for later use. The key point is to ensure the objectives are described adequately in the catalogued description of the analysis. For example, the role of "driving an M1 tank in road convoy" as required to support logistics or battlefield surveillance objectives will lead to a different set of simulation requirements than the role of "driving an M1 tank in road convoy" as required to support a scenario involving the response to an air attack. In the first case, the requirement might be met by a rather unsophisticated automated (i.e. SAFOR) simulation, whereas in the later case a sophisticated manned simulator might be required.

Attempting to just catalogue the role "driving an M1 tank" would tend to result in overspecification, because the role so broadly defined would have to include the most demanding type of mission which involves driving a tank. This does suggest, however, that if one is willing to accept the risk of overspecification, then roles might be defined hierarchically. For example, "driving an M1 tank" might have subcategories "for driver training" and "for tactical scenarios" and so forth.

One could not use such categories blindly, as there is always the possibility that the new simulation requires a role not envisioned in what was previously thought to exhaust the category. For example, the new role might involve a combination of weather conditions and obscurants on a particular type of terrain that had not previously been encountered. Such possibilities make it unlikely that any category as broad as "all tactical missions" could be supported by a complete list of requirements.

Simulators can also be catalogued according to their capabilities. Such a categorization would not be concise, as it takes a hundred or more parameters just to completely describe a simulator visual system. Once categorized, it would thereafter be easy for a computer to match simulators with requirements. The software could also provide "closest match" information, suggesting, for example, that a particular simulator would meet a requirement if only a certain visual database were ported to it.

In sum, much of the analysis needed to validate a networked simulation can be added by cataloging requirements according to roles and simulators according to requirements. This type of cataloging, built around analyzing roles, is very different from "equal simulator" cataloging, as pointed out below.

"LEVEL OF FIDELITY"

A proposal, originating from the Battlefield Distributed Simulation - Developmental (BDS-D) program, has been made [3] along the following lines:

1. The objectives of a simulation exercise will be related to a "level of fidelity."
2. Individual simulators will be assessed and assigned levels of fidelity.
3. For mixed levels of fidelity among simulators in an exercise, differentials of fidelity will be computed and held to within a tolerance allowable for the particular exercise.

This approach follows from the proponents assertion that "The key is that fidelity is determined by the accuracy and realism required for a given exercise." This proposal is therefore strongly at variance with the present contention that fidelity requirements should be derived for each simulator based upon the role the individual player has in the exercise.

To attach a useful measure of fidelity to an exercise as a whole, the fidelity of each of the simulators involved must be reasonably tightly grouped. This is implied by the notion of a tolerance placed on the allowable differential. This notion does not therefore seem to apply to the situation where a single exercise encompasses roles requiring vastly different levels of fidelity. Single exercises can, however, include roles with widely varying requirements for fidelity, and as exercises tend to reflect the diversity of roles of a real battlefield, the concept of having a nearly common level of fidelity becomes unsustainable.

Consider the visual simulation requirements for a single exercise having a battlefield surveillance system like JSTARS, non-line-of-sight weapons, support units, armor units, and attack helicopters. The surveillance system requires no visual simulation at all and the attack helicopter requires a highly sophisticated visual simulation, and

the other units have requirements somewhere in between, each appropriate to its role in the exercise. One could attempt to assign an overall average level of fidelity to the exercise, but the tolerance from simulator-to-simulator would have to be so large as to invalidate the concept. Under no circumstances could we ignore the individual roles, as it would make no sense to assign a visual system to a unit like JSTARS that has no need at all for one.

Note that even within a single armored vehicle, the fidelity requirements may vary considerably. A tank driver has a relatively near horizon, and has a limited field of view, but needs good detail in nearby objects. The commander has a much wider field-of-view and needs good detail in the distance, but little detail nearby.

This observation leads to a second difficulty with any simple measure of level-of-fidelity: there are far too many parameters related to level-of-fidelity to admit a simple measure. Just comparing the roles of commander and driver in the same vehicle, with little analysis we immediately found a number of parameters needed to characterize the differences in requirements, including differing distributions of visual objects. For widely diverse roles the number of parameters will be in the hundreds. Military systems use all manner of radio and microwave emission sensors, visual and infrared sensors, acoustic sensors, and even sensors that sniff the air.

Mapping the large number of independent parameters into just a few is pointless, because doing so conceals the individual requirements we need. For systems that sniff the air, the ability to sense the atmosphere is vital to their function, to other systems that ability is not relevant, so no fixed weight could be assigned to that ability in an overall measure of fidelity.

Even when weights can be assigned, the problem remains. For example, suppose we attach weights to near-detail, far-detail, and field-of-view to compute an overall measure of fidelity. We then compute that the driver needs a level of fidelity that is 37 on a scale of 100, whereas the commander requires a level of 46 on the same scale. Providing simulators that meet the total fidelity measure without correctly apportioning the individual requirements makes no sense.

The concept of level-of-fidelity may be of some use in the case where all the roles in an exercise happen to be the same. Such scenarios do occur, and may be common in SIMNET-type applications to date. However, cases where roles are all similar are the easiest to handle with an analysis relating role requirements to individual simulator requirements. Such an analysis would seem to be required in any case to establish the tolerance allowed on the differential in fidelity. The level-of-fidelity concept therefore seems to have only a narrow application which should be carefully considered in the light of the greater scope of the problem.

SUMMARY

The basic means of validating a network simulation is to identify the roles of each player and to make sure that the requirements of each role are matched by the simulator assigned for the exercise. The fidelity requirements for different players may vary dramatically, but so long as each player can carry out the role assigned, the overall objectives will be met.

The methods for analyzing roles to translate them into requirements have been well established for procuring individual training systems. Cataloging the analysis data may simplify applying analysis to individual networked simulation exercises.

Methods which do not take into account the differences in the requirements for different roles are limited to situations in which all the roles are similar, otherwise they will lead to gross over-specification of requirements for the less demanding roles.

REFERENCES

1. Dees, J.W., and Cornett, T.R., "Simulator Networking in Helicopter Air-to-Air Combat Training," AIAA Flight Simulation Technologies Conference, Boston, MA, September, 1989

2. Latham, R.W., The Use of an Environment Server Function to Improve Interoperability in Networked Simulation, Proceedings of the 14th Interservice/Industry Training Systems Conference, San Antonio, TX, November, 1992, pp. 26-36

3. Woodard, P., et. al., "Measuring Fidelity Differential in Simulator Networks," Proceedings of the 14th Interservice /Industry Training Systems Conference, San Antonio, TX, November, 1992, pp. 713-717

The 1992 I/ITSEC Distributed Interactive Simulation Interoperability Demonstration

Margaret Loper, Brian Goldiez, and Scott Smith
Institute for Simulation and Training
University of Central Florida
Orlando, FL 32826

ABSTRACT

The first demonstration of the Distributed Interactive Simulation (DIS) Protocol Data Unit (PDU) standard was conducted at the 14th Interservice/Industry Training Systems and Education Conference (I/ITSEC) in San Antonio, Texas in November 1992. This effort was sponsored by the Defense Modeling and Simulation Office (DMSO) and the US Army's Simulation, Training and Instrumentation Command (STRICOM).

The DIS standard protocol data units (PDU) and current communications architecture were utilized along with the common visual databases using Project 2851 (P2851) data. The demonstration was an integrated display of both standardization efforts. The Institute for Simulation and Training (IST) at the University of Central Florida developed the detailed design of the demonstration system, coordinated the effort for the government, and provided technical support to those organizations who demonstrated interoperability at the I/ITSEC. Planning Research Corporation (PRC), the P2851 contractor, prepared the databases.

This paper describes the approach used and lessons learned from the interoperability demonstration. The planning and integration effort consisted of three components. First, the scope of the demonstration had to be determined. This included three main issues: the communications network, the DIS standard, and the terrain database. Second, before integration occurred, each simulator had to be tested for compliance with the DIS standard. The testing was conducted at the San Antonio Convention Center during the week prior to the I/ITSEC Conference. The last component of the effort was the scenario developed for the opening plenary and banquet demonstrations. The scenario was dependent on the outcome of testing and was therefore the most dynamic component of the effort.

ABOUT THE AUTHORS

Ms. Loper received a B.S. degree in Electrical Engineering from Clemson University in 1985 and a M.S. degree in Computer Engineering from the University of Central Florida in 1991. She is currently Principal Investigator for the Distributed Interactive Simulation (DIS) project at the Institute for Simulation and

Training, Orlando. Her current research interests include simulation networking, OSI protocols, and multicast communications.

Mr. Goldiez is the Director, Research and Development at the Institute for Simulation and
Training at the University of Central. His research interests include modeling and analysis of dynamic systems, systems testing methods, and verification, validation, and accreditation methodologies. He received a B.S. in Aerospace Engineering from University of Kansas and a M.S. in Computer Engineering from the University of Central Florida.

Mr. Smith received a M.S. degree in Computer Science from the University of Central Florida in 1981. He has more than 14 years of experience in Software Engineering and Operating Systems design. He has worked in product development in both commercial and defense sides of the computer industry. At IST he has served as Principal Investigator on a research project investigating the generation of automated forces in a networked simulation environment. He is currently managing several projects related to the development of Distributed Interactive Simulation. His areas of interest include modeling of human behavior in real-time simulations, Aggregate-level wargames, and application of advanced computer architectures to DIS.

The 1992 I/ITSEC Distributed Interactive Simulation Interoperability Demonstration

Margaret Loper, Brian Goldiez, and Scott Smith
Institute for Simulation and Training
University of Central Florida
Orlando, FL 32826

INTRODUCTION

In March 1992, the concept for a real-time demonstration of the Distributed Interactive Simulation (DIS) standard, known as DIS 1.0, was conceived for the 14th Interservice/ Industry Training Systems And Education Conference (I/ITSEC) held in San Antonio, Texas on 2-5 November 1992. The demonstration was held with concurrence of the sponsoring I/ITSEC organization, the US Air Force, and was sponsored by the Defense Modeling and Simulation Office (DMSO) and the US Army's Simulation Training and Instrumentation Command (STRICOM).

The DIS standard protocol data units (PDU) and current communications architecture were utilized along with the common visual databases using Project 2851 (P2851) data. The demonstration was an integrated display of both standardization efforts. The Institute for Simulation and Training (IST) at the University of Central Florida coordinated the effort for the government and provided technical support to those organizations who demonstrated interoperability at the I/ITSEC. Planning Research Corporation (PRC), the P2851 contractor, prepared the databases.

This joint activity involved a wide variety of organizations. Each participant brought expertise in one or more aspects of the demonstration. In particular, IST developed selected portions of the demonstration system and also served as a clearing house for interested parties desiring more information, wishing to participate, or needing help with specific technical aspects of the effort.

SCOPE

Though the extent of what DIS can support is broad, the scope of the demonstration was restricted by the limited preparation time. The I/ITSEC demonstration was a joint application that utilized manned and unmanned simulated vehicles plus one live vehicle (not meeting DIS requirements). In addition to the manned and unmanned simulators, a few I/ITSEC demonstration participants simply "listened" to the network and used the information as input to radar simulations or to a "window" into the battle environment. The I/ITSEC application demonstrated the capability of heterogeneous simulations to interact in a common environment using the DIS protocol. The degree of correlation and the realism of the exercise was limited by the lack of experience with the standards.

The scope of the demonstration was defined by the participating companies through a set of planning meetings held at IST. At these meetings, issues pertaining to the network, DIS standard, and terrain representation were discussed and voted on. Issues which required further research before coming to a decision were taken as action items by IST, studied, and presented to the participants at the following meeting. All action items and decisions were documented in a report called "Actions and Decisions" which was distributed to all organizations participating in the planning meeting. The planning meetings took place over a period of seven months. In concert with several meetings, tutorials were held on different components of the demonstration.

General
Over the 8 month period, 28 organizations directly supported and/or participated in the planning meetings and demonstrations. There were a total of 18 manned and unmanned simulators, 22 "listen only" devices (network monitors, Stealths, etc.), and 1 live device used in the demonstration. This translated into 8 air simulators, 7 land simulators, 3 sea simulators, and 1 live land vehicle. Of the 18 manned and unmanned devices, 4 were Computer Generated Forces (CGF) systems. The organizations and types of simulators which participated in the demonstrations are shown in Table 1. In addition to simulator participation, the planning meetings and demonstrations were supported by STRICOM, USAF ASC, DMSO, PRC, Evans & Sutherland, and Star Technologies.

COMPANY NAME	TYPE OF SIMULATOR
BBN	Plan View Display, CGF, Stealth
CAE Link	AH-64, Stealth, Data Logger
Concurrent	Network Monitor
GD Air	F16
GD Land	M1
Grumman	E2C
Hughes	UAV, JSTARS
IBM/ECC	After Action Review, Battle Master, M1
IDA	Stealth, Data Logger, Plan View Display
IST	CGF, Network Monitor, Data Logger, Stealth
Lockheed-Sanders	TSAD, Scenario Monitor, Patriot
Loral/GE	M1 Tank, Live M1, Taper (Live), Plan View Display
McDonnell Douglas	F16/SAM Sites, Network Monitor
Motorola	Surface Ship
NRaD	LHD Surface Ship, Stealth
NTSC	F/A-18, Surface Ship
Reflectone	Radar
Rockwell	F16
SG/Mak Tech.	Stealth
TSI	Stealth

Table 1: I/ITSEC Demonstration Participants

The I/ITSEC participants spent a total of two weeks in Texas.

The first week, 26-31 October, was for testing and integrating the DIS simulators. Testing, performed by IST, included all aspects of networked simulation: communication protocols, DIS PDUs, terrain orientation, appearance, and interactivity.

The second week was the I/ITSEC Conference where two formal exercises were scheduled and presented. The first demonstration was presented during the opening session of the I/ITSEC Conference on Monday, 2 November 1993 in the Lila Cockrell Theater adjacent to the convention center exhibit hall. The second demonstration was given immediately before the I/ITSEC banquet on Tuesday, 3 November 1993. This demonstration was given in the exhibit hall on a screen erected directly over the IST booth located at one end of the hall. In addition to the formal demonstrations, the DIS network was available for use during regular conference hours. This time was divided into: 1) free play, where participants could get on the network and engage in non-scripted play with other people, and 2) 30 minute blocks, where participants could "own" the network and conduct an exercise of their choosing.

The participants decided in early planning meetings to make the network public. Any organization could play on the network as long as it did not interfere with any other player on the network. The decision to develop a mutually beneficial network was based on the philosophy of "demonstrating, not evaluating" the DIS Interoperability Network.

During both weeks, a voice communication network was established using contractor furnished walkie-talkies to provide a capability to control and coordinate the rehearsal play.

Network Design

The network design for the I/ITSEC demonstration consisted of two parts: one network for testing simulator interoperability during the eight months prior to the conference and another network for the actual DIS demonstration at the San Antonio Convention Center. Accordingly, the design of the network took place in two phases. The first phase included the design and implementation of a network at IST which allowed participants to test their DIS simulators against a system known to be DIS compliant. The second phase of development was the design of a network which supported the demonstration of DIS during the formal exercises, the free play, and the 30 minute time slots during the week of I/ITSEC. One issue which spanned both the IST network and the I/ITSEC network was the choice of communication protocols. Several options were available and the decision was based, in part, on the recommendation of the communication architecture for DIS (CADIS) draft standard being developed by the DIS workshops.

The choice of protocols for the I/ITSEC demonstration was decided by popular vote. At the initial March meeting,

participants made several proposals:

Layer	Possible Choices
Application	DIS
Network[1]	UDP/IP SIMNET Assoc. CLTP/CLNP Null
Link[2]	Ethernet IEEE 802.3

The OSI Connectionless Transport Protocol/Connectionless Network Protocol (CLTP/CLNP) was quickly eliminated as too new and too complex to implement for a near term demonstration, and a null network layer had little support. The SIMNET Association protocol was eliminated as being too closely associated with a particular company and product, whereas UDP/IP was an existing standard which could be purchased COTS.

A poll of the I/ITSEC participants at the May meeting showed a clear preference for Ethernet over IEEE 802.3, and so Ethernet was selected. Hence, I/ITSEC used a protocol stack of DIS/UDP/IP/Ethernet.

DIS Standard

The DIS standard used in the demonstration was Version 1.0 dated 8 May 1992. See Reference [1]. Version 1.0 of the standard covers a large scope of what DIS can support. Due to the limited preparation time, certain rules and restrictions were placed on the way this version of the standard was actually used. In addition to these restrictions, a set of policies was negotiated to determine the level of interoperability to be achieved.

The DIS standard defines a set of PDUs that achieve the basic requirements for distributed interactive simulation. Each PDU is divided into two fundamental parts: a mechanism and one or more policies. Mechanisms are static and are not changed. These are the PDU fields. For each PDU field, there are a variety of policies that may be applied. For example, in the Entity State PDU there is a field (mechanism) for a dead reckoning model. There are several dead reckoning algorithms (policies) that can be used. The policies used in the I/ITSEC demonstration were negotiated by participants during the planning meetings held at IST.

Only a subset of the PDUs listed in the DIS standard were used for the demonstration. These were the Entity State, Fire, Detonation, and Collision PDUs. Though the Collision PDU was part of the exercise, air entities were exempted from collision tests. This decision was based on a quick survey taken after 20 October when IST received a request from one of the participants that air entities be exempted from collision tests. IST contacted the air participants, upon which they unanimously agreed that collisions were not necessary for the I/ITSEC DIS demonstration.

Terrain Representation

The delivery of the terrain database was the responsibility of the P2851 team, a joint

project designed to develop common database formats. Vendors took the common data formats and converted the data into a form suitable for their computer image generators. Data from one vendor can be put into the P2851 format and be made available to other users. There are several formats available from P2851 which include the generic transform database (GTDB) format and the SSDB interchange format (SIF). SSDB refers to the Standard Simulator database which is the format P2851 uses internally. The SIF data format was selected for use by I/ITSEC participants.

The SIF database used for I/ITSEC was selected to be a 100 x 100 km area which included portions of Fort Hunter Liggett, CA. The geodetic coordinates of the southwest corner of this database were chosen to be N35-15-0, W122-4-0. Terrain, culture, and models were to be prepared for this area.

TESTING

The verification and validation of DIS compliant systems for the I/ITSEC demonstration were accomplished through the development of a testbed at IST. To make the testbed a reality, four key elements had to be developed: a test plan, a test system, test methods, and testing policies and procedures.

First, a test plan had to be developed which would serve as a guideline for testing simulator compliance with the DIS PDU standard. The test plan defined the interoperability requirements for participation in the DIS I/ITSEC interoperability demonstration. The level of interoperability defined was for the demonstration only and did not constitute conformance with the DIS standards for other applications. However, the test plan can be considered a subset of a full test implementation. The test plan was developed by IST over a period of four months and was then presented to demonstration participants for comment and review.

Second, a test system that was known to comply (by means of passing the test plan) with the DIS PDU standard was needed for organizations to test their DIS simulators against. This "golden system" had to be open and accessible to all participants who wanted to test their DIS simulators. The test system chosen was IST's Intelligent Simulated Forces CGF Testbed. Prior to testing, the CGF system underwent a conversion from SIMNET to DIS.

Test methods, the third element, were also important. How would demonstration participants access the test system at IST in order to test their systems against the test plan? Three economical and flexible alternatives were established which provided participants with a means to test via modem, data logger, or in-house. The modem method was only partially implemented.

The fourth element was the "Testing Policies and Procedures" document which established the ground rules IST followed throughout testing to ensure a fair and level playing field for all

organizations participating in the demonstration.

Minimal testing took place prior to I/ITSEC; therefore, the majority of all systems had to be tested once IST personnel arrived in Texas. During the first week, IST tested 41 systems in 84 hours, with all but one system passing the test plan. Desensitized test data and integration information is presented in [2]. By mutual agreement, each company's test results are confidential.

THE FORMAL DEMONSTRATION

IST developed the scenario for the formal demonstrations. The scenario was designed to provide a setting to demonstrate DIS interoperability and the capabilities of the participant's networked simulators without fear of intentional or inadvertent destruction by another player. To ensure a "win-only" scenario for demonstration participants, BBN's CGF system was used to provide opposing forces. They were not allowed to fight back and died when fired upon.

The control console was a Stealth or "magic carpet" which provides an "eyeball" view into the 3-D computer generated synthetic environment. The Stealth view was shown on the three center screens. This magic carpet was used to transport the audience to any point in the environment. The job of its operator was to give the audience the best view of the battle.

The scenario used for both formal demonstrations is described below:

(1) Two bogeys (SU-25s) were generated by BBN and detected by the E-2C. One target was assigned to the USS Ticonderoga and the other was assigned to the F/A-18 Combat Air Patrol.

(2) The first ship seen was the USS Wasp. It was generated in the NRaD booth. The NRaD ship had the ability to display any airborne or surface threat on its radar display by capturing location data from the DIS network.

(3) The second ship seen was the USS Perry and was generated in the Motorola booth. The Motorola ship also had the ability to display any airborne or surface threat on its Tactical Plot, as well as to launch missiles against these threats.

(4) The third ship seen was the USS Ticonderoga, generated in the NTSC booth. The NTSC ship also had the ability to display any airborne or surface threat on its SPA-25G radar and tactical plot.

(5) The first bogey came within range. The *Weapons Free* command was given to the USS Ticonderoga. The Stealth was used to show results of the firing of the missile from the ships and aircraft.

(6) Two F/A-18s were directed by the E-2C to intercept and destroy the second

bogey. The *Weapons Loose* command was given to the F/A-18s. The lead F/A-18 was generated in the NTSC booth.

(7) The second F/A-18 was generated in the Rockwell booth in the exhibit hall, but the pilot was physically located at the Rockwell plant in Los Angeles. The locations of targets and friendlies on the DIS network were being sent from the Rockwell booth via land lines to the domed simulator in California. The pilot flew his aircraft in response to these images and the resulting aircraft locations were transmitted back to the booth and into the DIS network for others to see and interact with. The Stealth was used to show results of the firing of the missile from the lead aircraft.

(8) The scenario play then jumped inland to view the land forces in the Hunter Liggett area. To save time the Stealth was attached for a ride on CAE Link's Apache helicopter. The Apache flew North at over 100 knots into the engagement area at Fort Hunter Ligget.

(9) The first unit seen was a Patriot Detachment generated in the Lockheed Sanders booth. The Patriot simulator had the ability to display, acquire, and engage air threats on the DIS network.

(10) The Patriot Radar picked up two approaching enemy attack aircraft on their display and the command was given to the Patriot battery, *You Have Permission to Fire*. As the Patriot battery was overflown, the Stealth was detached from the AH-64 to allow the audience to watch as the missiles were launched. The enemy aircraft were CGF entities generated in the McDonnell Douglas booth. The Apache continued north and spotted two enemy tanks (also CGF entities) generated by BBN. The Apache helicopter was given the command, *You Have Permission to Fire*. The Stealth was used to spot the action and the Hellfire missile firings.

(11) The next places visited were the battle positions of Task Force Alamo which was responsible for the defense of a critical road junction. As the Stealth approached the Task Force, a total of four tanks were exposed. Two of the tanks were seen off the right side of the road. An M1A1 tank was deployed forward in a fixed observation position in support of the dismounted infantry to their front. The tank was generated in the IBM booth.

(12) The first M1A1 tank to be seen was generated in the Loral booth. Two more M1A2 tanks which were

seen on the right side of the road were generated in the General Dynamics Land Systems booth.

(13) Placed well forward of the vehicle positions was a dismounted infantry (DI) fireteam. It was located to cover a route of advance not visible from the vehicle positions. This DI fireteam was generated by the IST CGF Testbed.

(14) Just ahead of the DI fireteam was seen the first of many Opposing Force (OPFOR) vehicles generated by the BBN CGF system in their booth in the exhibit hall.

(15) The IST DI fireteam was ready to engage the lead enemy vehicle with a Dragon missile. The DI fireteam was given the command, *Permission to Fire*. The audience watched as the DI kneeled, aimed, and fired the Dragon, destroying the lead OPFOR vehicle.

(16) The Stealth operator was commanded to rejoin the tanks in their battle positions to watch as the battle unfolded. The Task force was given the command *Permission to Fire*. The M1 simulators engaged the OPFOR with direct fire.

(17) An unmanned aerial vehicle was sent into the battle area. The UAV was generated in the Hughes booth. The UAV was assigned to fly through enemy held territory and to transmit simulated real-time TV sensor visual data back to the commander. The commander, seeing an advancing enemy armored column, called for close air support.

(18) An F-16 was generated in the General Dynamics, Ft. Worth booth and was flown from a simulated F-16 cockpit. The F-16 was tasked to engage an enemy mobile missile vehicle, a SAM. The SAM was generated in the McDonnell Douglas booth.

ISSUES AND RECOMMENDATIONS

Several systems level factors are important to consider when configuring and testing simulators and networks which are going to be integrated into a DIS environment. These factors include: minimizing the number of new technologies which are going to be integrated (i.e., P2851 and DIS), assessing simulator and network capabilities during the design phase (and not the implementation phase), avoiding the use of partial or reduced scope tests, testing ALL aspects of the design, having back-up designs which have been tested prior to implementation, and having sufficient time and support mechanisms in place to conduct necessary tests. Each of these areas will be further expanded below.

Combining the prototype products for the first time presents difficulties which should be avoided. Such was the case with P2851 and DIS.

Neither project had running prototypes for the I/ITSEC. The difficulty in the case of I/ITSEC came during integration. It was impossible to determine if a problem was due to terrain mis-correlation or misuse of DIS. For example, floating tanks in a visual scene could be the result of incorrect coordinate transformations, incorrect dead reckoning, or correlation problems between differently rendered databases. The causes of such situations are impossible to determine from I/ITSEC data. In the future, prototype products should be evaluated prior to integration with other system elements.

Simulator and network capabilities should be assessed during the design phase. In the case of I/ITSEC, the simulator and network capabilities were determined when the system was implemented during the rehearsal period. Part of the reason for the lack of information was the lack of validated tools to assess network performance given certain simulator and network characteristics. The second reason for the lack of information was an unwillingness by participants to assess or provide information on their simulators' capabilities. IST believes the lack of simulator information was due to the participants' lack of a firm commitment to the I/ITSEC hardware and proprietary considerations. The development of network assessment tools useful to simulation's needs will solve part of the problem. A willingness to share information or to make non-disclosure agreements will solve proprietary information problems.

Partial test procedures should be avoided. Interoperability was achieved at the I/ITSEC partially by leaving details of the scenario open until just prior to the demonstration. The need was partially due to not using detailed test procedures. I/ITSEC participants did not have time (or probably budget) available to develop special software specifically for testing. IST's detailed test procedures required simulators to perform in ways for which they were not originally designed. For example, IST may have asked simulators to pitch up 90° in order to check Euler angles and proper interpretation of rotation commands. These rotations were to be performed at the center of the earth to separate translation from rotation problems. A tank simulator may not have had such a capability. This problem can be avoided if testing procedures are standardized resulting in one time development of test software.

All aspects of the simulator network design should be tested. IST did very little testing of simulators under conditions involving adverse or erroneous data. In addition very few network performance tests were conducted. IST should have conducted performance tests of the various components of its own testbed and the integrated testbed system performance. Such tests would have resulted in better data gathering capabilities.

Backup designs which have been tested are important to one time demonstrations. The network problems just prior to the start of I/ITSEC have been documented. Something similar to a "failure modes and effects analysis" should be conducted in advance to anticipate problems and determine spare requirements.

Sufficient time should be planned into development efforts or demonstrations. There was insufficient time available to design, build, and test the simulation network at I/ITSEC. The demonstration was successful, in part, because the audience had no expectation of what was going to be demonstrated and the scenario could be adjusted to accommodate the special needs of simulators and the network. Future demonstrations or integration efforts must have realistic time budgets, if for no other reason than audiences now have an expectation of DIS and P2851 capabilities and are going to expect ever increasing sophistication of simulator networking.

CONCLUSIONS

Demonstrations can be useful if properly structured. The DIS demonstration served to show technology advancements and the utility of simulator networking to a wide group of interested parties. The DIS demonstration also served as an excellent example of technology transfer. Companies worked together to arrive at common understandings and solutions to interoperability problems. This helps to guide the development of standards and testing methods. Demonstrations should set out clear goals and show how those goals have been met. Demonstrations should be used as a means to collect data. The DIS demonstration at I/ITSEC was the first instance of data collection for simulator networking which was made available to the public.

The DIS protocols work. DIS is a robust set of protocols which have a wide range of applications. However there are several cautions which must be observed in using DIS. First, DIS is still developmental. The various versions of DIS are not compatible with each other making interoperability difficult. It is hoped that the emergence of DIS 2.0 will provide a stable baseline for product development and system implementation. Second, the DIS standards provide a wide range of options to users. The options must be selected for each instance of simulator interoperability. Third, the PDU level of standards is incomplete for interoperability. Definition of the environment (or methods to assess simulated environment similarities) is necessary for interoperability. Fourth, sound testing methods are necessary for DIS conformance and interoperability. Finally, the DIS Steering Committee needs to carefully manage the proliferation of DIS. Uncontrolled proliferation of PDUs or arbitrary control of new ideas could restrict the applicability of DIS.

REFERENCES

[1] Military Standard (Draft), Protocol Data

Units for Entity Information and Entity Interaction in a Distributed Interactive Simulation, Version 1.0, IST-PD-90-2 (revised), 8 May 1991.

[2] I/ITSEC DIS Interoperability Demonstration Test Procedures and Results, M. Loper, B. Goldiez, S. Smith, H. Ng, M. Craft, M. Petty, G. Bulumulle, and C. Lisle, IST-TR-93-04, February 16, 1993.

ACKNOWLEDGEMENT

This work has been supported by the U.S. Army's Simulation, Training and Instrumentation Command (STRICOM) under contract N61339-91-C-0103. The views and conclusions in this document are those of the authors and do not represent the official policies of the funding agency, the Institute for Simulation and Training, or the University of Central Florida.

1. The Transport and Network Layers are combined as "network."

2. The Data link and Physical layers are collectively called "link."

Reviewing the Battle at the Alamo

Grace C. Mak-Cheng
Kenneth Doris
Grumman Corporation
Great River, New York

Robert Perry
Norm Lawler
Concurrent Computer Corporation

ABSTRACT

The creation of the synthetic, virtual battlefield at the 14th I/ITSEC in San Antonio demonstrated the feasibility of the use of the non proprietary Distributed Interactive Simulation (DIS) protocols for the interoperability of dissimilar simulations. Although a major milestone has been reached in the demonstration of the ability of dissimilar simulators to communicate with the DIS protocol, true interoperability has yet to be determined. The actual interoperability of the players cannot be assessed until a thorough review the individual player's action and response has been made.

During the demonstration, a data logger developed by Concurrent Computer Corporation was used to collect all message traffic on the DIS network. Grumman, in conjunction with Concurrent, has begun a post mission review of the data collected. This paper will describe the findings of this review. A comparison of how the actual network traffic compared with the predicted assumptions, and how the use of the next order dead reckoning algorithms may impact the network traffic will be made. Discrepancies as a result of differences in the terrain database and interpretations of the rules of engagement will be pointed out. This paper will also include the "lessons learned" from this review process.

ABOUT THE AUTHORS

Grace Mak-Cheng is an Engineering Specialist in Grumman's Combat Systems organization. She is currently responsible for an Independent Research and Development project involving real-time networking of distributed simulation. She actively participates on the Standards for the Interoperability of Defense Simulations. Ms. Mak-Cheng is the principal investigator for Grumman's development of a DIS Network Interface Unit. During the 14th I/ITSEC, this unit was used to network Grumman's Flight Instrument Trainer to the DIS network for the DIS Demonstration. She holds a Masters in Computer Science from New York Institute of Technology and a Bachelor of Electrical Engineering from New York Polytechnic Institute of Technology.

Kenneth Doris is a Technical Advisor in Grumman's Combat Systems organization. He is currently directing several research projects, including one devoted to DIS investigation. He is an active member of both the Communications Architecture and the Emissions subgroups of DIS. Last fall Mr. Doris lead the Grumman team at the 14th I/ITSEC DIS demonstration held in San Antonio. He holds a Bachelor of Electrical Engineering from Rensselaer Polytechnic Institute and has twenty-five years experience in Simulation and C^3I, specializing in computer architecture and software engineering

Robert Perry is a member of Concurrent Computers' Professional Services group. He works on a consulting basis for Concurrent Computer Corporation customers, as well as in-house special projects. During the 14th I/ITSEC, he authored a DIS monitor which recorded and graphically displayed all network activity. Mr. Perry holds a Bachelor of Electrical Engineering from George

Washington University and has eighteen years experience in the simulation and computer sciences.

Norm Lawler is a member of Concurrent Computers' Engineering division. He is a project manager responsible for a number of Internal Research and Development projects involving distributed computing technology, specifically client/server application development for real-time environments. He actively participates on the Standards for the Interoperability of Defense Standards. Mr. Lawler holds a Bachelor of Science from University of Western Australia and has eleven years experience in networked applications development.

Reviewing the Battle at the Alamo

Grace C. Mak-Cheng
Kenneth Doris
Grumman Corporation
Great River, New York

Robert Perry
Norm Lawler
Concurrent Computer Corporation

INTRODUCTION

The concept for the real time demonstration of the Distributed Interactive Simulation (DIS) Standard at the 14th Interservice/Industry Systems Education Conference (I/ITSEC), which was held in San Antonio, Texas November 2-5, 1992 was conceived only eight months prior to the demonstration. Although the extent of what DIS can support is broad, the scope of the demonstration was restricted by the limited preparation time. During the eight month period, the twenty-eight organizations which participated in the demonstration worked together to define the scope of the demonstration.

The I/ITSEC demonstration was an integrated display of two standardization efforts: the DIS standard protocol data units (PDU) and communications architecture, and Project 2851, the standard for common visual data bases. The Institute for Simulation and Training (IST) at the University of Central Florida coordinated the testing and integration efforts. The pre-demonstration testing included testing of the communication protocols, DIS PDU's, terrain orientation, appearance and interactivity. Although participants were required to pass the test administered by IST before participation in the demonstration, as this paper will point out, it was possible to pass the test and not be DIS compliant.

For the demonstration, Concurrent Computer Corporation developed a network monitor which logged all the messages which were sent on the DIS network. This paper will discuss the development and implementation of the network monitor. This paper will also describe the analysis of the data collected during the demonstration. The findings of these analyzes may aid in the further development of the DIS compliant test bed, the development of future data loggers and network monitors and development of a better understanding of the DIS standards.

I/ITSEC DIS Demonstration Ground Rules

The I/ITSEC demonstration was a joint application of manned and unmanned simulated vehicles. A few of the I/ITSEC demonstration participants "listened" to the network and used the information as input to their radar simulations or displayed a "window" into the simulated battlefield environment.

DIS Network

The participants decided to make the DIS network public. This meant that anyone could play on the network as long as he or she did not interfere with any other player on the network. The participants used IP broadcast directed to UDP port 3000 (decimal) for legitimate DIS traffic. Any non-DIS messages put on the network during the demonstration were to be sent point-to-point if possible, and if not possible, by multicast. Each company was assigned 10 unique UDP port numbers for non-DIS traffic.

DIS PDU's & Dead Reckoning

The DIS standard used in the I/ITSEC DIS demonstration was Version 1.0 dated May 8, 1991. Only a subset of the PDU's listed in the DIS standard was used for the demonstration. These PDU's were the Entity State, Fire,

Detonation, and Collision PDU's. Though the Collision PDU was part of the exercises, air entities were exempted from collision tests.

With the Entity State PDU, a relative time stamp was used as a result of the absence of a global network timing mechanism. Articulation parameters were only used on some of the ground based vehicles. Of the 64 bits in the articulation parameter, the first 32 bits were used to indicate the turret azimuth and gun elevation. The remaining 32 bits were padded with zeros.

With the Detonation PDU, no articulated parameters were present in the PDU since no damage models were used in the DIS demonstration. Damage assessment models were excluded to reduce the complexity of the exercise.

The dead reckoning model used was the first degree model. The threshold parameters for issuance of new Entity State PDU's were three degrees and one cubic meter.

Terrain Database

The delivery of the terrain data base was the responsibility of the Project 2851 team. Vendors took the common database formats and converted the data into a form suitable for their computer image generators. The data base was distributed to participants in SSDB (Standard Simulator Data Base) interchange format (SIF). The data base selected for the I/ITSEC demonstration was a 100x100 km area which included portions of Fort Hunter Liggett, CA. Although there were some known discontinuities in culture and terrain, the tight schedule made freezing the data base necessary. A high resolution area of 10 km N/S and 30 km E/W was specified as the area containing all ground vehicle activity. Participants were advised to convert the high detail area as faithfully as possible. The error threshold requested by participants was set to 1.0 meters.

Testing

The IST's test plan defined the interoperability requirements for participation in the DIS I/ITSEC interoperability demonstration. The level of interoperability defined was for demonstration only and did not constitute conformance with the DIS standard for other applications. However, the test plan can be considered as a subset of a full test implementation. Details of the test plan, test procedure and test results have been published by IST (Ref. 1).

Development of the DIS Monitor

The monitor was developed on a Concurrent Computer model 7100 computer. This machine included three Motorola 68040 processors, 32 Mbytes of memory, a graphics display system (GA-5000), and integral Ethernet and SCSI controllers. This hardware was driven by Concurrent's Real Time UNIX operating system (RTU), X-Windows, and the DIS monitor application.

The goal of the DIS monitor was to provide a real-time visual display of network traffic on a per player basis as well as log all network activity to disk for later review. The demands of this goal required that the application utilize the real time extensions to UNIX that RTU provides. This includes priority scheduling, CPU dedication, and very efficient inter-process communication mechanisms. The monitor application was based on four co-operating (UNIX) processes: a net reader, a statistician, a disk writer, and the display system. Each of these processes attached to a shared memory area and coordinated all data access through locked counters and asynchronous traps.

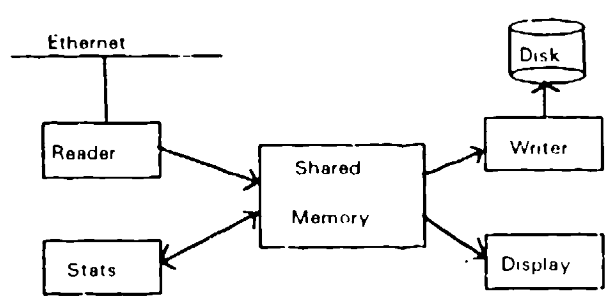


Figure 1 Concurrent Computer Corporation Network Monitor

Net Reader - This process used the Data Link Programming Interface (DLPI) available in UNIX. DLPI allows an application to bypass all network protocol software thereby providing the most efficient access to the net while still

maintaining hardware independence. In addition DLPI passes the "raw" Ethernet frame to the application. Each frame includes hardware source and destination addresses. These addresses proved very useful in identifying players during an exercise and in simulating all players during a replay operation. The net reader issues reads on the net and when data is available, it time stamps the packet, and loops waiting for the next read completion.

Statistician - This process reviews data provided by the Net Reader and decodes each packet according to type on both a player and net basis. This process runs until it has processed all the new data provided by the Net Reader, at which time it suspends execution.

Disk Writer - This process logs <u>all</u> packet data and associated time stamps to disk. Similar to the Statistician, this process continues to execute as long as new data is available.

Display - This X Windows based program graphically displays the output of the Statistician process using custom bar chart and strip chart Widgets. There are two screens in the monitor. One shows overall network activity while the other indicates the type of activity by the specified player.

As the packets are read from the net, each is time stamped using the internal gettimeofday() UNIX system call. For analysis, this stamping procedure is very important as there was no synchronized time base across the network. It is assumed that by using DPLI, real time priorities, and the pre-emptive kernel of RTU, any delay associated with packet transmissions and reception within the host machine is constant. One "problem" discovered during the analysis regarded the interpretation of the recorded time stamp. The time recorded is based on GMT while the standard UNIX routine that interprets this time takes into the account the time zone of the machine doing the analysis.

POST MISSION REVIEW

The post mission review involved examining two issues: (1) Network Traffic, and (2) Entity Interactions. Analysis of the network traffic involved examining the number of packets which was issued by each entity and the network bandwidth consumption of these packets. A second order dead reckoning algorithm was applied to the Entity State PDUs issued by each entity to determine the effect of this higher order algorithm on the network traffic. Since only four DIS PDUs were used at the demonstration, the entity interaction analysis was limited to examining the Fire/Detonation event sequences and the Collision detection.

ACTUAL NETWORK TRAFFIC

Network Packet Analysis

During the 14th I/ITSEC demonstration, Concurrent Computer Corporation made a number of logs of the network traffic, which included some test sessions as well as the plenary session. The following analysis is based on the data log of the second plenary session which was recorded on Tuesday, November 3, 1992, for one hour starting at 5:30 p.m. This period of time was chosen since it represents a pattern of network traffic usage under a "controlled" DIS scenario and is not distorted by DIS testing and debugging activities.

Two key aspects of networking which affect DIS are the processing load imposed on each host on the network by DIS traffic, and the consumption of the available network bandwidth. The load on each host system is directly related to the number of network packets which have to be processed, where each packet imposes an interrupt and processing overhead before the DIS data can be made available to the simulation application. The bandwidth issue is related to the amount of data transmitted over the network, typically in terms of bits per second (bps).

Out of the total of 150,000 network packets logged during the analysis period, 96,334 were DIS PDUs and 53,666 were non-DIS packets. Figures 2 and 3 show a further breakdown of the DIS PDUs and the non-DIS packets:

	Entity State	Fire	Detonation	Collision
No. of PDUs	96,213	61	56	4

Figure 2 DIS PDU Types

	ARP	ICMP	TCP	OTHER UDP
No. of Packets	5025	48,364	149	128

Figure 3 Non-DIS PDU Types

As expected, Entity State PDUs make up the bulk of the DIS traffic. As all of the DIS PDUs were UDP broadcast packets, every host should have received and processed all 96,334 PDUs. Averaged over the one hour analysis period, this represents a per host load of only 27 PDUs/second. A more detailed analysis showed that the worst case sustained load occurred over an 18 second interval during which the average load was 112 PDUs/second, with a peak of 139 PDUs/second. During this interval 28 entities contributed to the network traffic, six Anti-Armor Maverick guided missiles produced 78% of the total Entity State PDUs during this time.

A total of 98 simulation entities was recorded over this one hour period. The breakdown of the number of PDUs issued by each entity type is shown in Figure 4.

The large number of non-DIS packets, particularly the ICMP (Internet Control Message Protocol) and to a lesser extent the ARP (Address Resolution Protocol) message, was unexpected. The ICMP packets were almost totally of Type 3, Code 1 ("host unreachable"), all from the same source, indicating that the probable cause was one particular host on the network used a non-conformant IP protocol suite. This host sent ICMP error messages as a result of receiving a datagram defined to an IP broadcast address. (See Section 3.2.2 of RFC 1122 - Requirements for Internet Hosts -- Communications Layers).

Interestingly, there was a period of approximately 27 minutes during the middle of the one hour analysis period when almost no ICMP messages were logged, indicating that the problematic host was either powered down or disconnected from the network at this time. Since every ICMP message was directed to a specific host, this should not have imposed any extra unnecessary processing load on the other hosts on the network.

Network Bandwidth Analysis

The 150,000 network packets (DIS and non-DIS) transmitted during the one hour analysis period resulted in 21.4 Mb of data being sent over the network. This gives an average bandwidth usage of only 0.0475 Mbps over the one hour period. The peak bandwidth consumption for this one hour demonstration only reached 0.15 Mbps, thus using only 1.5% of the available 10 Mbps bandwidth capacity of the Ethernet. Moreover, the 1.5% bandwidth usage was a result of at least one ICMP packet being transmitted by the errant host discussed earlier. Further analysis showed that only 16 simulation entities contributed to the peak loading.

Figure 5 shows the rates (per minute) at which DIS and non-DIS packets were transmitted over the network. The peak for both types of traffic during the second minute is the point of peak bandwidth usage, while the DIS PDU peak in minute 28 was the point of maximum DIS PDU load for the hosts on the network. This graph shows the close correlation between the DIS broadcast PDUs and the ICMP messages discussed earlier.

The total number of DIS-related and non-DIS-related bytes (summing the complete Ethernet frame sizes contained in the packets) transmitted over the network came to 17.75 Mb and 3.65 Mb respectively.

Dead Reckoning Algorithm Analysis

For the I/ITSEC demonstration, the first degree dead reckoning model was used. Since the fields in the Entity State PDU which would allow for higher orders of dead reckoning (i.e., the acceleration and angular velocities) were in most cases not filled in by the participants, a detailed analysis of the use of other dead reckoning algorithms was not possible.

KIND/ DOMAIN	NUMBER IN EXERCISE	ENTITY STATE PDU	FIRE PDU	DETONATION PDU	COLLISION PDU	% OF BANDWIDTH
PLATFORM/ LAND	52	15739	6	6	2	16
PLATFORM/ AIR	16	71065	31	27	0	75
PLATFORM/ SURFACE	4	2237	23	23	2	2
MUNITION/ ANTI-AIR	14	4294	0	0	0	4
MUNITION/ A-ARMOR	6	1935	0	0	0	2
MUNITION/ A-SHIP	5	924	0	0	0	1
LIFEFORM/ LAND	1	19	0	0	0	0
TOTAL	98	96,213	61	56	4	

Figure 4 DIS PDU Breakdown

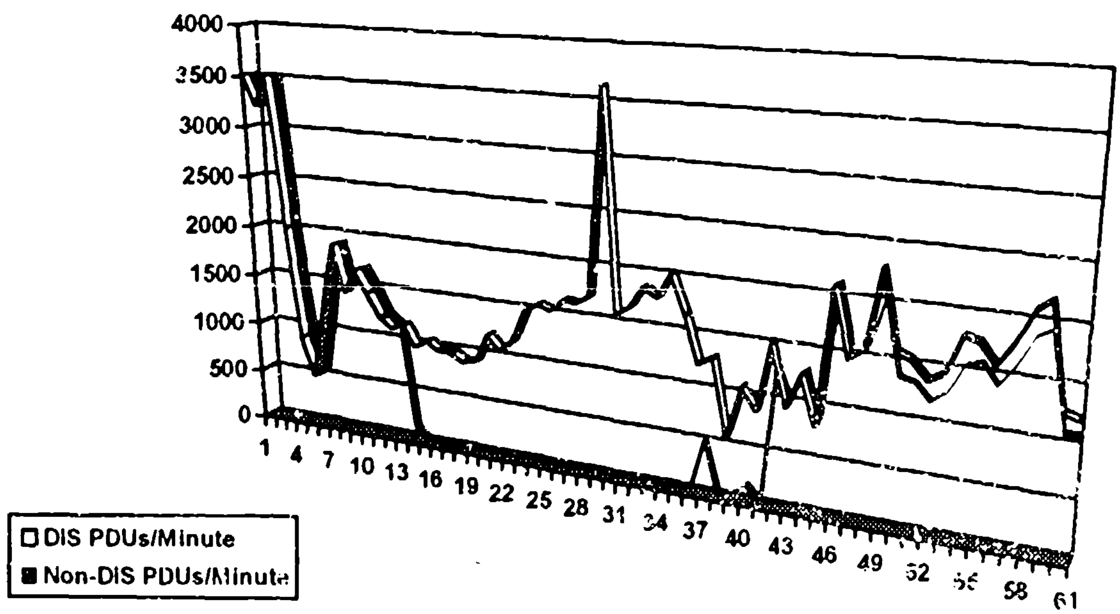


Figure 5 Network Bandwidth Analysis

I/ITSEC EXERCISE TRAFFIC ESTIMATES							
% ENTITIES AT HIGH RATE		0%	20%	40%	60%	80%	100%
% ENTITIES AT LOW RATE		100%	80%	60%	40%	20%	0%
# of TANKS	52 →	17,638	51,551	85,463	119,375	153,288	187,200
# of AIRCRAFT	16 →	4,608	41,216	77,824	114,432	151,040	187,648
# of SHIPS	4 →	1,152	2,240	3,328	4,416	5,504	6,592
# of TACTICAL VOICE LINKS	0 →	0	0	0	0	0	0
# of TACTICAL DATA LINKS	0 →	0	0	0	0	0	0
TOTAL TRAFFIC BITS/SEC	→	23,398	95,007	166,615	238,223	309,832	381,440
PDU/SEC	→	14	62	109	156	203	250

However, it was possible to analyze the effects of a second degree dead reckoning model with the recorded data by calculating the entity's acceleration from two frames of velocity data. A new position was calculated with the acceleration and velocity terms. Using the recorded data as the "truth" for the entity's position, a comparison was made to determine if an Entity State PDU needed to be issued as a result of exceeding the threshold limits of one cubic meter.

The results of this analysis showed that the use of a second degree of dead reckoning resulted in a saving of 1634 Entity State PDUs. The uses of a second degree dead reckoning algorithm resulted in no savings of Entity State PDUs for entities which were land platforms (i.e., tanks); however, a saving of as much as 17% was seen with some of the air platforms. With the entities which were munitions, little savings were seen (i.e., only a saving varying from one to five Entity State PDUs).

Predicted Network Traffic

The predicted network traffic is based on a network bandwidth analysis program which was written by Grumman (see Ref. 2). Figure 6 shows the predicted network traffic for the given number of entities which were recorded during the plenary session. The observed average rate of 27 PDUs/sec matches well with a predicted rate of about 10% activity while the peak observed rate of 139 PDUs/sec would match about 50% activity. The predicted traffic in bits/sec of approximately 200K bits/sec, however, is somewhat high in comparison to the observed peak of 150K bits/sec.

Discrepancies in entity interactions

This portion of the analysis is still ongoing. One observation that can be made is that not every Fire PDU was followed by a corresponding Detonation PDU (61 vs. 56). The next step will be to look in detail at the time difference between the issuance of the Fire PDUs and the associated Detonation PDUs, and to examine whether the intended targets match in both.

Lessons Learned

The data logs included a header which contained the exact time in seconds, from January 1, 1970, GMT, in which the data logging began and when it finished. Each logged Ethernet frame was similarly time stamped. Unfortunately, the relevant timezone and daylight saving information (w.r.t the machine recording the data) were not logged which made it extremely difficult to match up data logs with known events which occurred at the specific times during the I/ITSEC demonstration (e.g., the start of the plenary sessions).

Suggestions

If the data logs are large, then a general requirement is to provide a way to analyze specific sections of a data log, and wall clock times are the most obvious means of identifying significant points in the log. If the analysis is to be carried out on systems other than the machine which recorded the data, then the timezone and daylight saving information must be recorded somewhere in the data log.

At the time of the I/ITSEC demonstration, no analysis tools had been implemented. While Concurrent Computer Corporation also had a Network Monitor displaying real-time the network traffic loads, it was not obvious at the time that ICMP messages were being transmitted over the network at such significant rates. It was only during later when analyzed over a reasonable time frame that ICMP messages were seen as significant and likely to cause a potential network problem. The lesson to learn here, is that analysis tools should be an integral part of any data logger and it should be possible to use the tools even during the data recording session to obtain a more complete view of network traffic patterns and be able to pinpoint network problems earlier.

References

[1] I/ITSEC DIS Interoperability Demonstration Test Procedures and Results; Institute for Simulation and Training, University of Central Florida; Feb. 16, 1993.

[2] Computer Architecture for Distributed Interactive Simulation (CADIS); *Militery-Standard (Draft)*; Institute for Simulation and Training, University of Central Florida; June 28, 1993.

[3] RFC 1122 - Requirements for Internet Hosts -- Communication Layer.

SIMULATION NETWORKING AT KIRTLAND AFB

Mark Castle, Kevin Curley, John Little, and Frank Magee
Martin Marietta Information Systems Company, Flight Systems
Daytona Beach, Florida, U.S.A.
1 June 1993

ABSTRACT

Under the sponsorship and direction of the 542d Crew Training Wing at Kirtland AFB and the Department of the Air Force Headquarters, Ogden Air Logistics Center (AFMC) at Hill AFB, Utah, Martin Marietta has implemented a real-time network for multi-device interactive simulation. Currently this network is on contract to interface the following air crew training devices and facilities: MH-53J Weapon System Trainer (WST)/Mission Rehearsal System (MRS), MH-60G WST, TH-53A Operational Flight Trainer (OFT), and the 542d Training Observation Center (TOC). The network designated SOF-NET, was integrated and ready for training (RFT) in 1993. In the near future, the network will expand to include the HC-130P, MH-60G OFT, Aerial Gunner and Scanner Simulator (AGSS), and an external Distributed Interactive Simulation (DIS) network node. The external node will be used to link the SOF-NET with other Government networks and facilities. To date, the MH-60G, MH-53J, and TH-53A helicopter simulators have been successfully tested for network interactions; in support of an accident investigation, key information was provided through a networked simulation of a multiple ship mission.

This paper examines the Kirtland network architecture and the implementation approach which links the varied computational platforms, Image Generators, Radar and EW Systems. The SOF-NET results to date and potential future projects suggest that this facility is a pathfinder site for the resolution of several thorny DIS issues such as data base correlation, EW simulation, virtual/ constructive interfaces and aggregation/deaggregation. The successful resolution of these issues as applied at Kirtland AFB may impact future revisions of the DIS specification and provide a basis for future interactive network applications.

ABOUT THE AUTHORS

Mark Castle is the Martin Marietta Program Manager for networks at the Kirtland AFB 542d Crew Training Wing facility. In this position he is responsible for integrating the existing SOF-NET into national networks. He has held positions in aerospace electronic systems design for the past fourteen years. Prior to his current position, Mark was Program Manager and Chief Engineer for an SDIO air defense projectile interceptor program. Mark holds a BS in Electrical Engineering from the Lawrence Institute of Technology and a MS in Engineering from the University of Pennsylvania.

Kevin Curley is Martin Marietta's Project Engineer for the Training Observation Center (TOC) at the Kirtland AFB 542d Crew Training Wing facility. He is responsible for the technical functionality of the TOC as it provides overall command and control environments for the four fully functional Weapons Systems Trainers utilizing the SOF-NET at the 542d. Previous to being assigned to Mission Support Systems Programs, Kevin was a hardware design and development engineer and lead systems engineer in large, parallel processing, environments for several major programs. Kevin holds a BS in Electrical Engineering from the University of Florida and a MS in Engineering from the University of Pennsylvania.

John Little is a Chief Engineer working with Business Development for Martin Marietta's Mission Support Systems Programs in Daytona Beach, Florida. He is currently responsible for developing solutions to technical challenges for new and current flight simulation programs. John's previous position was Chief Engineer for the MH-60G Weapon System Trainer (WST) program, under which the SOF-NET system was originally installed in the 542d training facility. John holds a BS in Computer Science from Arkansas State University.

Frank Magee is the Systems Engineer developing DIS application requirements and various inter-simulation demonstration scenarios. Frank worked extensively on the 542d's local Inter-Simulator Network requirements and design control effort. Frank, a former Marine officer, holds bachelor's and master's degrees in mathematics from Temple University and Villanova University, respectively.

SIMULATION NETWORKING AT KIRTLAND AFB

Mark Castle, Kevin Curley, John Little, and Frank Magee
Martin Marietta Information Systems Company, Flight Systems
Daytona Beach, Florida, U.S.A.

INTRODUCTION

An unprecedented capability in the high fidelity arena of helicopter crew training and mission rehearsal has been developed at the 542d Crew Training Wing located at Kirtland AFB. This capability centers around three helicopter training devices: the MH-60G Weapon System Trainer (WST), the MH-53J WST and the TH-53A Operational Flight Trainer (OFT). These trainers include full fidelity simulation of subsystems such as Forward Looking Infrared (FLIR), Digital Radar Landmass (DRLMS), Night Vision Goggles (NVG), Electronic Warfare (EW), and a fully realistic cockpit with an Out-The-Window (OTW) display driven by an eight-channel COMPU-SCENE V Image Generator (IG). The training capability of this facility has been further enhanced by the development of a state-of-the-art Data Base Generation System (DBGS) which currently provides high fidelity, fully correlated (OTW visuals with FLIR with NVG with radar and EW) data bases for each of the trainers. This combination of fully realistic training devices coupled with high fidelity data base production has enabled a unique training and mission rehearsal capability in support of SOF missions for individual crews.

This training/mission rehearsal capability has been significantly expanded with the introduction of the SOF-NET network, as shown in Figure 1. This network will allow SOF teams to train and rehearse for multiple ship missions. The hub of the SOF-NET network is the Training Observation Center (TOC). The TOC is a multi-media center which supports role playing, review, and replay of networked training and mission rehearsal exercises.

Future network expansion will support team training in joint exercises against highly realistic threats provided by other DoD facilities. This paper summarizes (1) design considerations for the SOF-NET, (2) the TOC, (3) the SOF-NET hardware implementation, (4) the SOF-NET software implementation, and (5) future SOF-NET expansions.

SOF-NET DESIGN CONSIDERATIONS

The training devices linked via SOF-NET are the MH-53J WST, MH-60G WST, TH-53A OFT, and the TOC. These devices represent a variety of computational platforms and a distributed processing environment which shaped SOF-NET's architecture and implementation. Figure 2 shows a block diagram representative of a typical helicopter trainer at Kirtland AFB. The block diagram emphasizes the major computational platforms including the SOF-NET interface. The specific computational platforms for each trainer are listed in Table 1.

The table illustrates that while the IG and basic instructor-operator systems are identical for the three systems, the host computational platform and underlying software approach for each of the systems are significantly different. Therefore, the key issue for the SOF-NET network architecture was to provide a common system capable of linking disparate trainer host systems. This posed unique challenges for software data structure designs and hardware interfaces between varying host computational platforms and a single network structure.

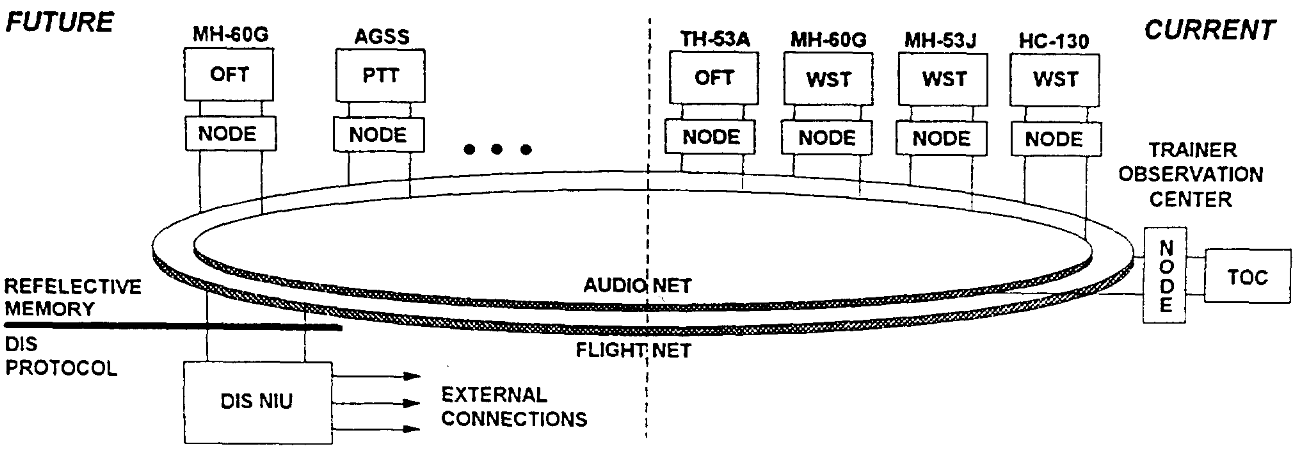


Figure 1. 542d CTW SOF-NET With Future Expansions

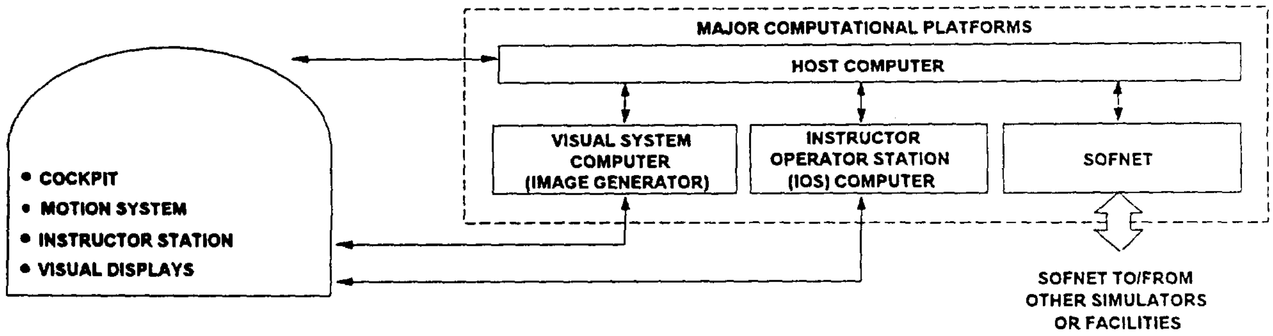


Figure 2. Typical SOF-NET Integrated Trainer Block Diagram

Training Platform Commonality

Each training system was developed with differing technology levels along with different contractors and differing design philosophies. While there were many issues to resolve in networking these trainers, the most fundamental issues were:

1) Basic host iteration rates
2) Intercommunication system capability
3) Host software fidelity

Intercommunication System Capability

The intercommunication systems construction varied widely, along with the flexibility and capabilities of each system. The TH-53A contains a strictly analog system from the 1970's, while the MH-53J and MH-60G both contain digital systems. The digital systems were different between the MH-60G and MH-53J, however, until late 1992 when a modification was placed in the MH-53J to adapt a system similar to the MH-60G onto the simulator. The interface design between the SOF-NET and the simulator had to be generic enough to allow for integration of a common design to interface with each of the three audio designs.

Table 1. Kirtland Trainer Major Computer Platforms

MAJOR CPU	MH-60G	MH-53J	TH-53A
HOST	Force Harris (DRLMS)	Encore Multi-Sel	Harris 500 w/Force Interface
IOS	Silicon Graphics	Silicon Graphics	Silicon Graphics
IG	Encore COMPU-SCENE V	Encore COMPU-SCENE V	Encore COMPU-SCENE V
ISN	Force	Force	Force

Basic Host Iteration Rates

For this element of the problem, the TH-53A executes at 16 Hz, the MH-53J executes at 60 Hz and the MH-60G executes at 30 Hz. Each simulator had its own timing scheme, and own system hardware limitations. This made the challenge of integrating a standard network with the hosts a real challenge. The basic approach is to update the network at high rates. This drove the requirement for a high bandwidth network which was implemented with the SCRAMNet™ reflective memory architecture.

SCRAMNet is a registered trademark of SYSTRAN Corporation.

Host Software Fidelity

The simulation system is only as good as its hardware and software components, and the diversity between the three simulators varied widely. Along with the differing iteration rates comes the differing levels of fidelity in the calculations associated with the visual interface, flight equations and instructor interfaces. Each of the three devices, being of differing heritage, had a different slant on the same problem. For example, the TH-53A is compiled in Harris Assembler, while

the MH-60G and MH-53J are written in Fortran. Here was the largest challenge, and where the most emphasis has been placed to balance between commonality in the network interface units, while minimizing modifications to the host systems.

Level of Fidelity

As the designs for each WST and the TOC have evolved, the requirements for the SOF-NET have matured and stabilized. The key SOF-NET design goals were to support the highest levels of fidelity while minimizing modifications to the individual WST's. The focus was to allow for two elements to be fully supported: 1) Visual correlation and 2) audio integration. Any peculiar element associated with network operation which was unique to a device was left to that device to either enhance or ignore. The SOF-NET integration requirement has evolved into the support of an interface which can generally be met by almost all simulators on the market today along with a direct correlation with the DIS standard.

Trainer interaction became a simplified set of primitives: one set for each trainer. Each data set contains all that is required to be known about that entity on the network. There have been some simplifications due to the fact that the helicopters which have been dealt with so far do not emit items such as missles or gun fire. (The SOF-NET will be updated to support these types of entities when the AGSS is integrated onto the network.) The data supplied in the entity blocks is sufficient to drive moving models and special effects for representation of the information on the visual systems of each of the WST's. Along with the ownship information, each entity block also contains all model information for any moving models driven by that ownship.

TRAINING OBSERVATION CENTER (TOC)

The TOC, as shown schematically in Figure 3, is a facility designed specifically to support the Kirtland SOF-NET. The TOC has several missions: 1) provide an electronic classroom which supports multi-media academic training such as computer-based training (CBT) materials, and live or pre-recorded audio and visual data from the simulators operating on the SOF-NET, 2) provide the capability for interactive role playing during multi-ship training scenarios involving audio communications with each training device, 3) provide the capability for flight tracking utilizing a map-based situational display, and 4) provide the capability to selectively monitor simulator visual data.

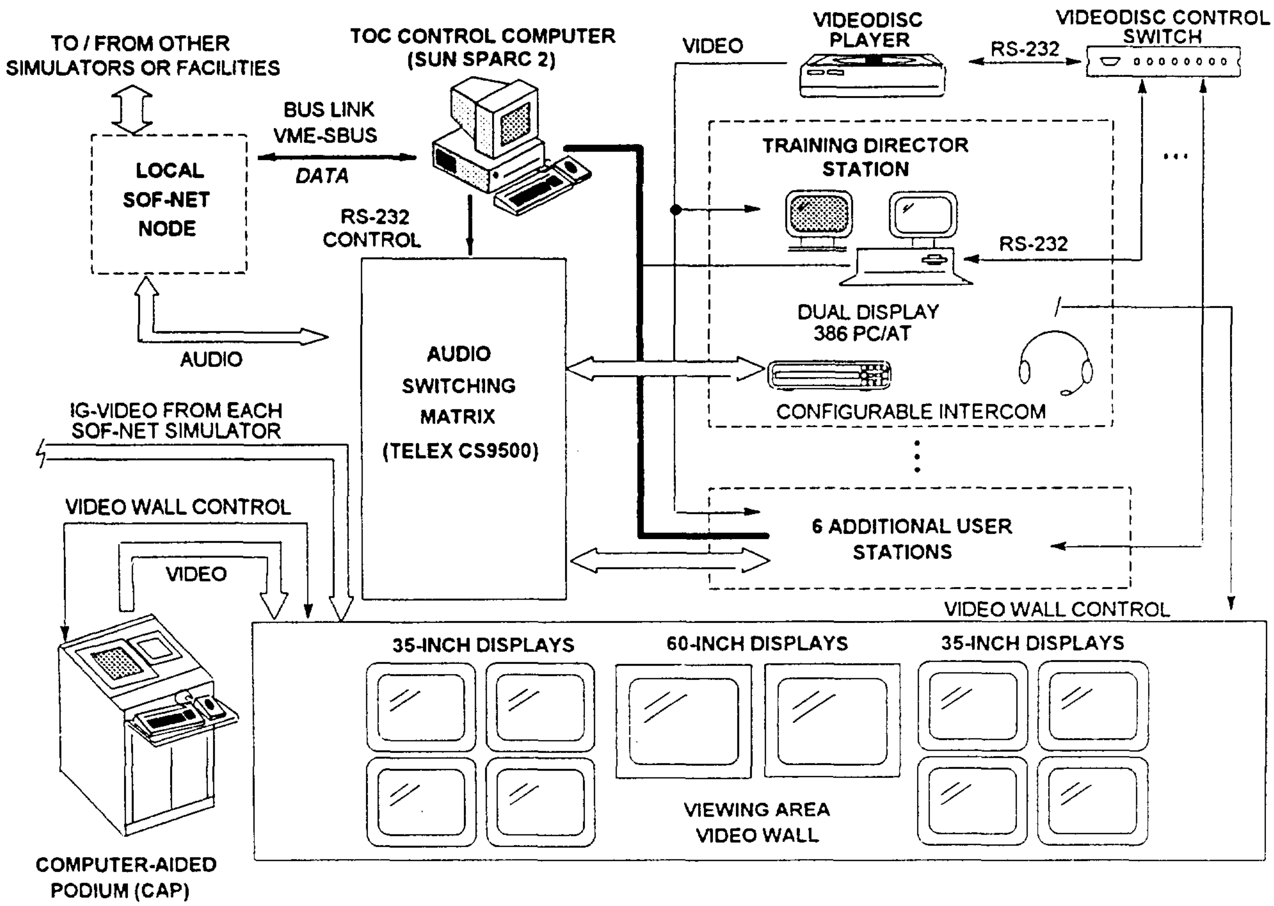


Figure 3. Training Observation Center (TOC) Block Diagram

Electronic Classroom

The TOC's electronic classroom capability centers around the Computer-Aided Podium (CAP). The CAP provides a user friendly, tailorable multi-media control interface. The CAP user can select a variety of visual options for video wall displays such as CBT, live video from any of the simulators on the SOF-NET, or pre-recorded video from a tape played on any one of the four VCR's available in the TOC. In addition, the TOC has an audio/visual recording capability which allows recording of any classroom or training activity conducted in the TOC for later review. The CAP is the control center for the TOC's electronic classroom.

Interactive Role Playing

The TOC has seven user stations each providing the capability for interactive role-playing during multi-ship SOF-NET missions. Each user has a 386PC/AT and a configurable intercom unit. The user PC provides a Microsoft Windows based interface for configuring the user's communications selections and accessing the Fulcrum situation display system. The user's intercom allows selection of any UHF, HF, or FM radio and frequency combination. Radio communications operate much the same as in the real-world in that the user will be connected for communications with any other TOC user and/or simulator crew position on the SOF-NET based upon radio/frequency matching. In addition to the three radios channels, each TOC user has a configurable instructor channel which allows private communications with a single simulator instructor or common connection to all instructors.

Situational Display

Fulcrum is a map-based mission tracking capability allowing the use of Videodisc or CD-ROM based maps. The user's Fulcrum situation display is automatically fed overlay data extracted from the SOF-NET by the SparcStation 2 acting as the TOC control computer. The extracted data consists of pertinent model, position, and support information for each player on the network. The user selects the desired map resolution and geographic region of interest. Fulcrum filters the overlay data based upon the user's selections and displays icons in the area being viewed (representing ownships, moving models, threats, and aircraft tracks). Fulcrum is operated by each user in a networked mode which allows the use of a single videodisc or CD-ROM drive to provide a map source while at the same time allowing each user to view the mission tailored to specific user interest.

The TOC control computer acts as the Ethernet LAN server, controls the audio switching matrix and provides a bridge to the SOF-NET data. The Ethernet LAN allows automatic distribution of SOF-NET situational data to each user station and access to user radio configuration information. The user radio configuration information is combined with simulator radio configuration data taken off the SOF-NET to configure the audio switching matrix.

Viewing Area Video Wall

The TOC video wall is configurable for a wide variety of display options. Video selections include VCR tapes, CBT video from the CAP, Out-The-Window (OTW) views from the IG of any of the SOF-NET connected simulators, NVG video or radar. The Fulcrum situation display from the Training Director's user station is selectable for viewing on one of the two 60-inch monitors allowing the TOC audience to monitor the multi-ship training mission. In addition, mission audio is selectable on the room speakers allowing the audience to fully track the mission.

SOF-NET HARDWARE

The SOF-NET is made up of multiple VME-based SOF-NET nodes and interconnecting cables (Figure 4). A node is defined as all hardware and software elements which provide an interface between the network and a single host training system. The hardware architecture of SOF-NET was driven by the two interfaces which exist at each node, 1) the host system interface consisting of data and audio, and 2) the network interface. In addition, security considerations required that each node be physically disconnectable from the network. As such, the hardware consist of five major functional components, 1) the Node CPU, 2) the network interface, 3) the host data interface, 4) the host audio interface, and 5) the optional EW system interface.

SOF-NET Node CPU

The SOF-NET Node CPU is actually a pair of Force CPU's, a Force 30 and Force 33. Two CPU's were needed to ensure time critical data transfers to/from the host could be met while still maintaining the required spare CPU capacity called out by the contract. The Force 33 acts as the VME bus arbiter and handles most of the calculation intensive computing such as interpolation of environments and distance and bearing calculations. The Force 30 acts as a real-time executive, provides an Ethernet interface used for booting both CPU's, and moves data between the host system and the SCRAMNet memory.

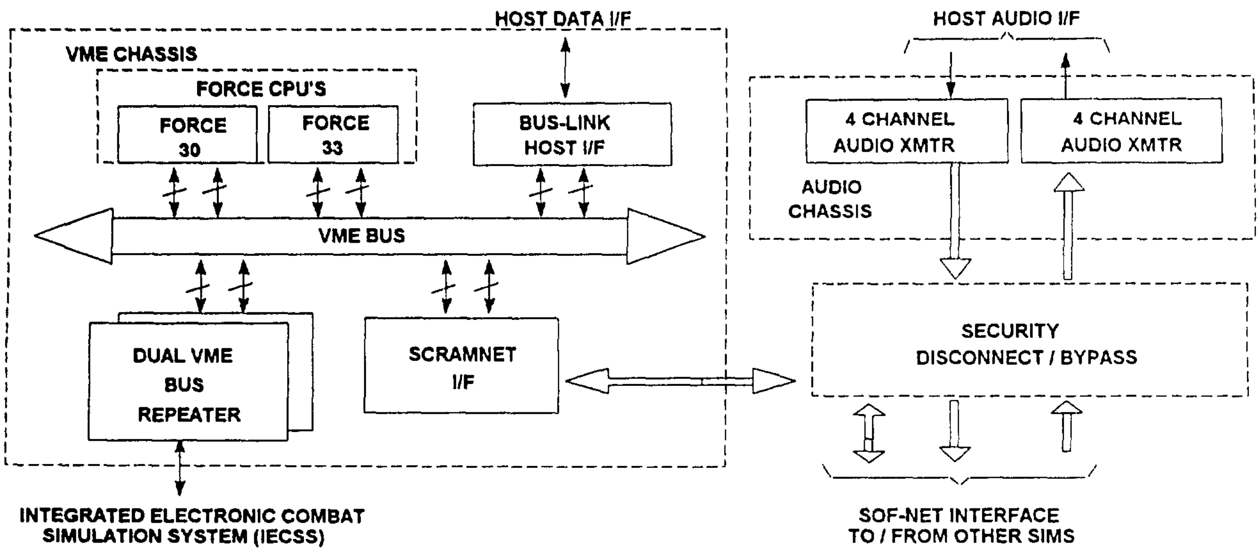


Figure 4. SOF-NET Node Hardware Block Diagram

Network Interface

The network interface functional component provides the means for each networked device to share data as rapidly as possible. This interface is facilitated with a reflected, shared memory scheme using SCRAMNet hardware provided by the SYSTRAN Corporation. The SCRAMNet board uses a fiber optic local area network configured as a ring, to provide high speed data transfers between each SOF-NET Node. Data is transferred across the network using deterministic processing and automatic retransmissions. The SCRAMNet on-board memory is parsed such that each training device is allocated a portion of the memory space which is "reflected" to all other nodes. At any given node, pertinent operating parameters and environmental data from the host training device are extracted and put into appropriate SCRAMNet memory locations. The SCRAMNet reflective-memory hardware makes this information available at each of the other SOF-NET nodes by sending it around the network.

Host Data Interface

The host data interface was selected to solve the need to minimize changes at each host system while satisfying a need for real-time access to the host system data pool. This interface was implemented using a Bus-Link subsystem which connects the SOF-NET Node VME bus directly to the Host system bus. The Bus-Link hardware selected is provided by Computer Products, Inc. (CPI). This connectivity allows the SOF-NET CPU access to a defined portion of the host system physical memory as a Remote Memory Interface (RMI). The SOF-NET CPU can move data in and out of the host memory as needed, allowing the host to operate independent of the SOF-NET connection while the SOF-NET Node CPU does the work to keep the host current with network mission model and environment data. The advantage of the Bus-Link is that the host side of the link is available in VME and Encore formats, allowing the implementation to be used at all three simulators. In addition, the bus-link technology was available for a VME-to-Sbus connectivity which was used in the TOC since the TOC has a Sun Microsystems SparcStation 2 as its control computer or host.

Host Audio Interface

The host audio interface was driven by the need to pass audio data securely over fiber optic cables and a desire to keep the implementation consistent with industry standards for line-level audio communications. The implementation uses COTS analog to optical audio equipment adhering to commercial standards which provide flexibility and the capability for future growth. Each SOF-NET node has a two board set of audio equipment except for the TOC. The TOC has a two board set for each of the other nodes. The overall SOF-NET audio architecture is a star with the TOC in the center. Each node passes audio to the TOC where it is mixed and routed according to audio configuration data received from each node via the SCRAMNet reflective memory. The TOC has an audio switching matrix which acts as the central switch for all SOF-NET audio. After audio is mixed according to the radio/frequency matching algorithm, appropriate audio is passed back to each node from the TOC. The result is broadcast quality audio in each of the radios and a flexible interface at each host which only requires compatibility with a balanced 2.2 Vptp audio input and output format.

EW System Interface

The SOF-NET interfaces externally with the Integrated Electronic Combat Simulation System (IECSS) built by TRW under subcontract to Martin Marietta for the MH-60G and MH-53J. This interface allows the EW system to actively monitor each player's movements and control threat to ownship interactions accordingly. The interface is implemented using two VME Bus repeater links. One link allows the EW system to access the SOF-NET node SCRAMNet memory without having to go through the node CPU and the second link allows direct access to the simulator host data pool via the SOF-NET node bus-link. The design minimizes the impact on both the simulator host and the SOF-NET node computers while facilitating the most direct and timely access to the needed data for the EW system. The EW access to the local host system allows control of the host on-board EW systems and threat encounters. Access to the SCRAMNet allows the EW system to use the SOF-NET as a means to pass pertinent EW data around the network. The result is a rich shared threat environment.

SOF-NET SOFTWARE

The software functions, distributed between Force CPU boards at each SOF-NET node, drive the SOF-NET's interactive simulator capability. Figure 5 depicts the SOF-NET software functional flow. Modular in design, this software is built for portability and commonality between SOF-NET nodes and executes asynchronously. SOF-NET operational software is structured into the following three independent CSC's.

1) Master/Slave
2) Network Moving Model/Control
3) Environmental Interpolation

All SOF-NET interfaces between the various SOF-NET CSC's are accomplished via shared memory. While the SOF-NET does share IECSS data between its various nodes, the software which writes/reads this data to /from the SCRAMNet resides on the hosts' IECSS's.

Master/Slave CSC

The Master/Slave CSC provides initialization, executive control, master/slave designation, ability of the master to select the common data base and control relocateable models. Relocateable model control allows the master to control the flow of model data to the network. This model data includes; navigation aids, ships, tanker aircraft, SOF teams, universal features, and IG special effects. The Master/Slave CSC player management function keeps account and control over the role being played by each simulator on SOF-NET, i.e., master or slave.

This CSC also provides for the implementation of other network management tasks. For example, the master player controls the environment of all trainers, while the slave player can control its own environment only when an override is selected. This CSC also calls the software units within the Environmental Interpolation CSC.

Network Moving Model Control CSC

The purpose of this CSC is twofold: 1) to transfer data associated with the host and its moving models to the network, and 2) to synthesize all data for active network moving models for the local simulator host. Each network host can donate its ownship and six other moving models to the network. This CSC extracts the appropriate data from the host data pool to sufficiently define the moving models to the network. The extracted data is inserted into the network reflected memory

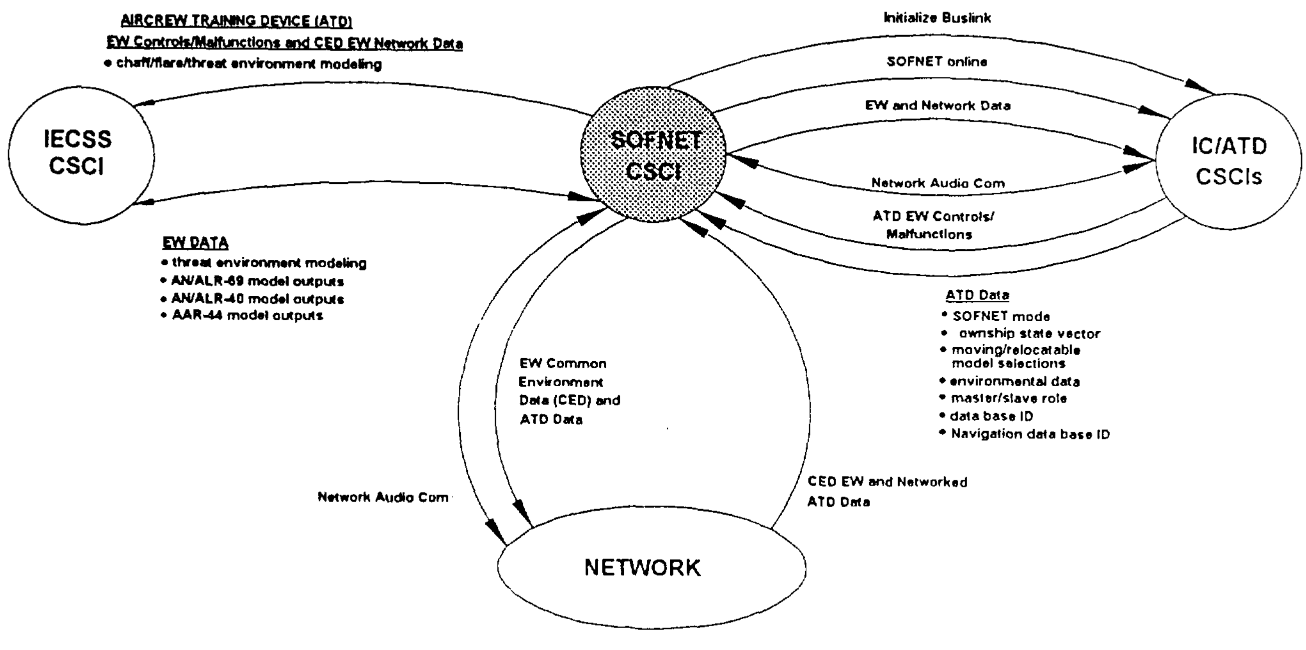


Figure 5. SOF-NET Functional Interfaces

space and subsequently available at all network nodes. Since each node can donate up to seven moving models to the network, each host must synthesize the model data to determine which models are most relevant to the ownship immediate environment. The synthesis algorithm is a sort by position relative to the host ownship. The 16 closest models are passed to the host computer for additional processing; the host will pass the six closest to its IG for display. The continual processing of all active network models ensures each player has the same model pool from which to build its visual environment and determine simulation impacts based on each host's unique capabilities for on-board systems such as EW, radar, and FLIR.

Environmental Interpolation CSC

The Environmental Interpolation CSC provides a means for a slave network player to smoothly transition into or out of the master environment by sharing the master's environment data across the network. As the slave passes within a distance threshold of the master's ownship position, the interpolation routine begins to linearly change the environmental state of the slave ownship to that of the master's. This interpolation is executed after the ownship comes on the net or after the slave comes out of environmental override. Some examples of the environmental state parameters are; precipitation, haze visibility, surface wind speed and other SOF/aviation related conditions.

FUTURE EXPANSION

In the near future, the SOF-NET network will be expanded to the nodes shown in Figure 1. The added nodes will include the MH-60G OFT (Operational Flight Trainer), the Aerial Gunner and Scanner Simulator (AGSS) and an external DIS Network Interface Unit.

The MH-60G OFT will support MH-60G Pave Hawk training. The OFT is a highly realistic non-motion simulator with seat shakers to provide motion cueing. The system will support day, night, dusk, and NVG flight operations in the existing high resolution data bases produced by the 542d's data base generation system (DBGS).

The AGSS is a part task trainer which will train SOF crew members in NVG scanning techniques and aerial gunnery for .50 caliber machine guns and 7.62mm mini guns. In the standalone mode, the AGSS will use a low cost small image generation system. When the AGSS is integrated into the SOF-NET network, the TH-53A OFT (COMPU-SCENE V) image generator will provide the visuals. This image generator switching will enable full crew operation between the AGSS and either the MH-53J WST or the MH-60G.

The eighth SOF-NET node is reserved for external Wide Area Network (WAN) integration with other simulation and simulator facilities. This node will be compliant with the design principles, goals and specifications promulgated by the current DIS (Distributed Interactive Simulation) protocol and future evolutions of the protocol. At this time, several government facilities and networks have expressed interest in linking with the SOF-NET network. The facilities of interest perform functions in two main areas: theater air defense with command and control nets and theater level constructive simulation for joint exercises.

The goal of the SOF-NET architecture is to provide a single node which will service all of these heterogeneous and diverse network connections. Effectively linking these simulation facilities with the SOF-NET network will focus on the challenges of integrating virtual man-in-the-loop simulators with a high degree of fidelity and high real-time update rates (as exemplified by the SOF simulators) with constructive upper level, nonreal-time wargaming simulations. A key aspect for the successful linkage of virtual and constructive simulations is the generation of a "common" data base for both levels of simulation.

Network Interface Unit (NIU) Implementation

The NIU will provide the connection to external facilities and networks. The implementation of the SOF-NET NIU is driven by the design guidelines of the DIS standard, as well as the implementation features of the individual WST's and the SOF-NET local area network. In the DIS architecture terminology ("Strawman Distributed Interactive Simulation Architecture Description Document," Volume I, 31 March 1992, ADST/WDL/TR-92-003010), the SOF-NET may be considered to be a cell of more or less homogeneous simulator entities connected by a network. In the two-tier model espoused by the strawman architecture, the SOF-NET NIU is considered to be a Cell Adapter Unit because it links a nonstandard cell of high fidelity simulators with a virtual inter-cell DIS network.

The NIU consists of a host computer, interface with the SOF-NET local area network, long haul communication equipment, and encryption devices. The choice of communications equipment and encryption devices will vary with specific network communications media and exercise security requirements.

A notional block diagram for a classified T1 connection is shown in Figure 6. In this application, a KG-94 acts as an encryption device. The Tl 12-slot chassis provides growth potential for additional Tl circuits and expanded audio capability. Another desirable feature of the unit shown is that it will support technology upgrades to T3 lines (45 Mbps) and Synchronous Optical Network (SONET) (at 50 Mbps to 2.4 Gbps). The outputs of the T1 chassis include four audio channels which are connected to the SOF-NET audio channels (HF, UHF, VHF, and IOS). The other output is a serial data link via Ethernet to the node host. The key requirements for the host processor are a real time, multitasking operating system, VME environment, SCRAMNet interface and growth potential for expanded network traffic, additional network connections and evolution of the DIS

protocol. The chassis shown contains a single CPU board with 40 MIPS throughput and 32 MB local memory, a SCRAMNet board and a Force 40 board for a COTS (commercial off the shelf) software package.

The NIU will be required to perform the following functions:

1) Translate data for the nonDIS LAN to/from DIS-compliant messages
2) Perform dead reckoning (remote entity approximation)
3) Encryption
4) Compression/decompression to facilitate network bandwidth requirements
5) Data shuffling to transfer data between reflective memory locations in the SCRAMNet architecture and the NIU memory buffer
6) Message filtering based on entity data and message content in order to prevent processing overload and minimize bandwidth requirements.

The software flow of these functions is shown in Figure 7. Our approach to the NIU has been to use third party vendor packages to the greatest possible extent. In the flow diagram, the functions with asterisks indicate functions which require application specific software. The selected third party package is the Advanced Interface Unit (AIU), developed by Naval Command and Control Oceans Surveillance Center RDT&E Division (NRAD) in conjunction with ETA Technologies Corporation.

Network Applications

Future plans call for integrating the SOF-NET with an advanced theater air defense facility. This facility provides a total integrated air defense simulation (both weapons and C3I) with several operator-in-the-loop, real time consoles. Its scenario capacity supports:

simultaneous tracks
emitters
ECM emitters
types of jammers
types of aircraft
terrain followers
controllable aircraft
active "threat" interceptors

This capability enhances SOF training for nap of the earth, stealth penetration into enemy air defenses. Also, communications systems such as TADIL-J will allow SOF teams to participate in command and control scenarios which may involve joint missions such as the location and defeat of critical mobile targets.

This SOF-NET project can serve as a testbed for several EW and tactical communications issues. The EW arena is driven by highly sophisticated, smart sensors and munitions which perform complex seeker, acquisition, tracking, jamming, and counter-jamming functions. High fidelity training exercises involving these types of systems will require high volume, high speed data transactions. These simulations will need to develop and incorporate sophisticated data compression techniques and imaginative preloaded data bases to limit network traffic to current state of the art bandwidth limits. Another issue which will be addressed is the issue of the shooter determining that he has scored a hit while the target registers a miss. These discrepancies can be caused by network latencies and by

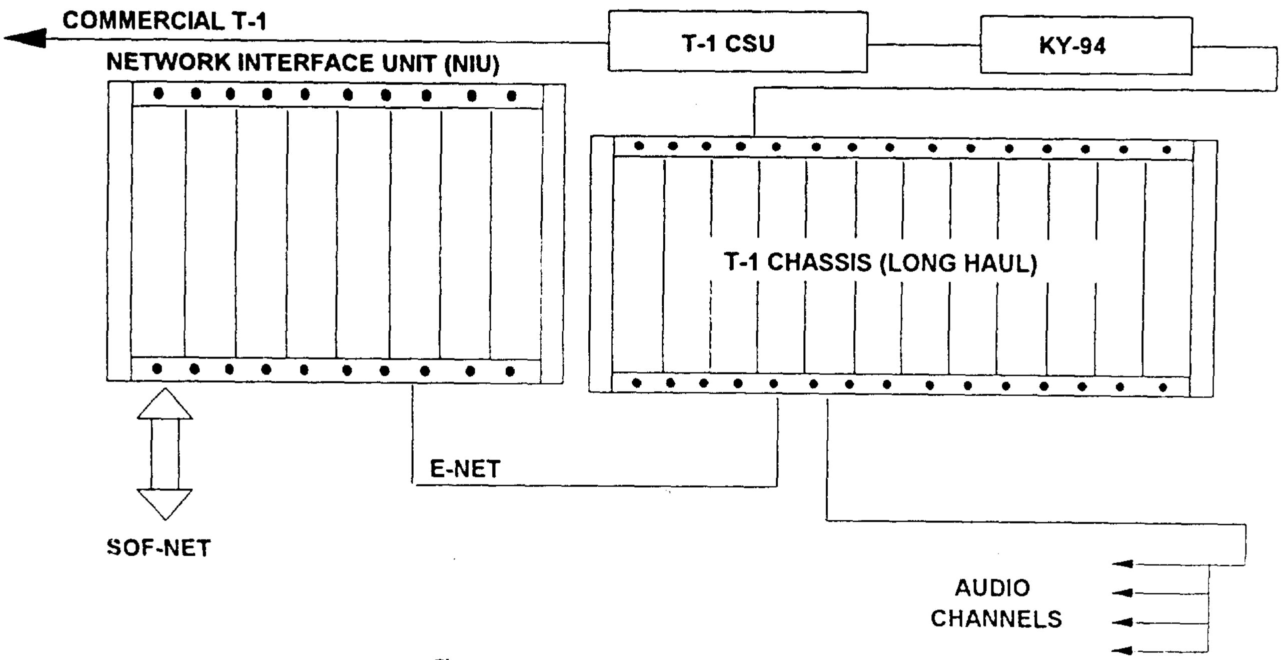


Figure 6. External NIU Block Diagram

improperly correlated EW data bases. These issues will be tackled by the proposed SOF-NET integration project. The resolution of these issues will provide guidance for the evolving DIS standards.

A second noteworthy future application is the linkage between SOF-NET and theater level constructive simulations. This effort will allow SOF-NET role players to participate in joint, theater level exercises. This SOF-NET integration will advance the state of the art in the area of virtual simulator/constructive simulation interfaces. The proposed integration effort will focus on the linkage of the SOF-NET with theater level simulations. This project will focus attention on the following DIS issues: Aggregation/deaggregation techniques between unit level simulations and platform (helicopter) level simulators; time coherence between faster than real time or asynchronous event driven simulations with real time simulators; DIS/ALSP protocol interfaces; and construction of correlated data bases for high fidelity, 3-D simulators playing in lower fidelity, 2-D simulation space.

CONCLUSION

The networking of the training devices via the SOF-NET has greatly enhanced the training and mission rehearsal capability of the 542d Crew Training Wing at Kirtland AFB. This training capability has expanded to coordinated team formation exercises. Future expansion to external facilities and networks will introduce training and mission rehearsal in joint service theater level exercises and in rich, man-in-the-loop threat environments. These applications will provide new insights to the interactive community and the evolving DIS protocol.

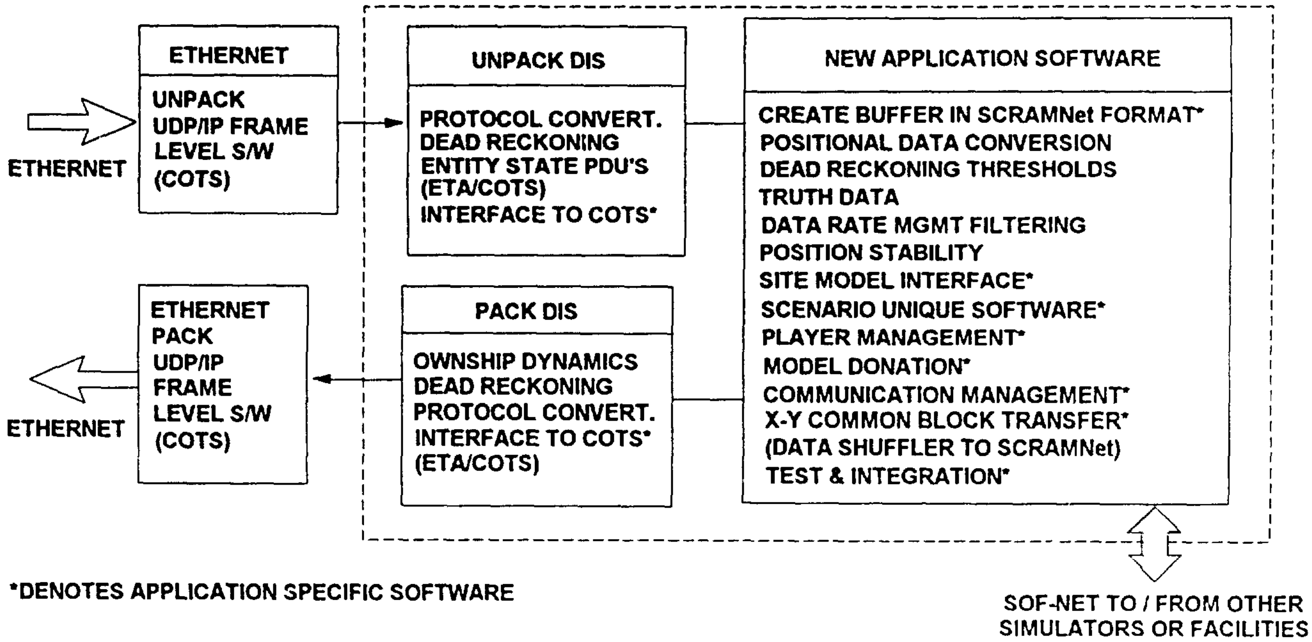


Figure 7. External Network Interface Unit (NIU) Block Diagram

Development, Test and Evaluation of a Multiship Simulation System for Air Combat Training

Capt Philip A. Platt, USAF Peter M. Crane, PhD
Air Force Armstrong Laboratory, Aircrew Training Research Division
Williams AFB, Arizona

ABSTRACT

The Aircrew Training Research Division of Armstrong Laboratory at Williams AFB, AZ has developed a SIMNET Version 6.6.1 compatible network of dissimilar aircrew training devices. The multiship research and development system (MultiRAD) uses distributed micro-processor technology to integrate: an exercise control and videotaping system, two high fidelity F-15 and two lower fidelity F-16 cockpits, visual display systems, a ground controlled intercept (GCI) station, and a computer generated threat system. As part of systems integration and development, four one-week tests were conducted in which F-15 pilots and air weapons controllers participated in simulated air combat training exercises using the MultiRAD system. During these exercises, pilots and controllers flew simulated offensive and defensive counter-air missions against a force of up to six threat aircraft plus surface-to-air missiles. Participants then evaluated the utility of the MultiRAD system for air combat training. System components were modified after each of the four weekly tests based on the participants' evaluations. Systems development, integration, and modifications, based on pilot and controller evaluations, are discussed along with lessons learned.

ABOUT THE AUTHORS

CAPTAIN PHILIP A. PLATT is a simulation and network research engineer assigned to the Aircrew Training Research Division of the Human Resources Directorate, Armstrong Laboratory, Williams AFB, AZ. He received his bachelor's degree in electrical engineering from Arizona State University in 1985 and a master's degree in computer engineering from the Air Force Institute of Technology (AFIT) in 1990. His previous assignment was to AFIT where he worked on low cost approaches to flight simulation. He is currently the lead computer engineer responsible for engineering and technical support for the Multiship Research and Development (MultiRAD) program.

PETER M. CRANE is a Research Psychologist with the Air Force Armstrong Laboratory, Human Resources Directorate, Aircrew Training Research Division at Williams AFB, Arizona. He received a BA in Psychology from Wittenberg University (1972) and an MA (1974) and PhD (1977) in Experimental Psychology from Miami University (Ohio). Before joining Armstrong Laboratory, Dr. Crane was an Assistant Professor of Psychology at the University of Pittsburgh at Johnstown. At Armstrong Laboratory, he has conducted research on the training effectiveness of simulated ground mapping radar imagery and simulated infrared imagery. Dr. Crane's current research is focused on identifying the training opportunities and instructional strategies for networked, multiple aircraft simulations.

Development, Test and Evaluation of a Multiship Simulation System for Air Combat Training

Capt Philip A. Platt, USAF   Peter M. Crane, PhD
Air Force Armstrong Laboratory, Aircrew Training Research Division
Williams AFB, Arizona

INTRODUCTION

The Advanced Research Project Agency (ARPA)/Army Simulation Network (SIMNET) program demonstrated that allowing combatants in simulators to interact with each other within a common gaming area greatly increased the value of simulator based training. Further, the SIMNET program demonstrated the feasibility of a network based on selective fidelity player stations using distributed microprocessor technology and an asynchronous communication protocol. Armstrong Laboratory's Aircrew Training Research Division evaluated the concept of networked simulator training for air-to-air combat in a series of advanced air combat exercises conducted at McDonnell Aircraft Company (Houck, Thomas, and Bell, 1989). Pilots in these exercises reported that the training received from networked simulators was superior to their current unit training for tasks which cannot be practiced in the actual aircraft due to cost, safety, and security restrictions. However, unlike SIMNET, the simulation facility at McDonnell Aircraft was designed for engineering development and uses very high fidelity cockpits and mainframe computer technology.

The multiship research and development program (MultiRAD) was initiated at the Aircrew Training Research Division in the spring of 1991 to create a SIMNET compatible system of networked simulators for air combat training. The initial objective of MultiRAD development was to integrate new and existing devices into a system which would provide high fidelity training for limited components of the F-15 air combat mission. The system would then be evaluated in a series of simulated air combat exercises known as the Training Requirements Utility Evaluation (TRUE). In the TRUE, teams of two F-15 pilots and an air weapons controller would either defend an air base against an attack or would escort a flight of F-16s attacking the air base. System performance and participant evaluations would then be used to identify the training opportunities provided by MultiRAD and to direct further MultiRAD development. In this paper, we will discuss the components in the MultiRAD system, the integrating network, a summary of the TRUE evaluation, and a discussion of lessons learned and opportunities for future development.

MULTISHIP RESEARCH AND DEVELOPMENT (MultiRAD) SYSTEM

The MultiRAD system consists of several independent systems connected via network interface units (NIUs) which convert each device's unique codes into a common communication protocol. The components used in the TRUE were: two F-15 cockpits, computer image generation and displays for the F-15s, two opposing forces cockpits, an air weapons controller station, a computer generated threat system, an exercise control and video recording station, and a separate video debriefing station (see Figure 1).

F-15 Cockpits

The two F-15C cockpits used were McDonnell Douglas Reconfigurable Cockpits (MDRCs). The MDRC incorporates high fidelity stick and throttle grips with a color CRT/touch screen depiction of the front panel (see Figure 2). The MDRC uses commercial, off-the-shelf, VME-based, microprocessors to perform all internal functions. Inside its single

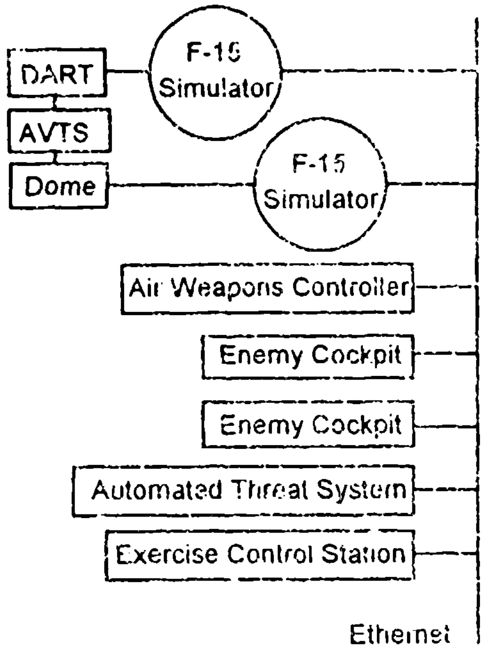


Figure 1. Multiship research and development (MultiRAD) network.

VME chassis, the MDRC has four Motorola 68030 single board computers, one Motorola quad-processor (86100) computer board, two computer image generator board sets, a sound board, and several digital to analog and analog to digital converters. The two image generators are used to provide the cockpit instruments and the head-up display (HUD). The sound board provides weapons cues and aircraft audio such as engine sounds and g-limit warnings.

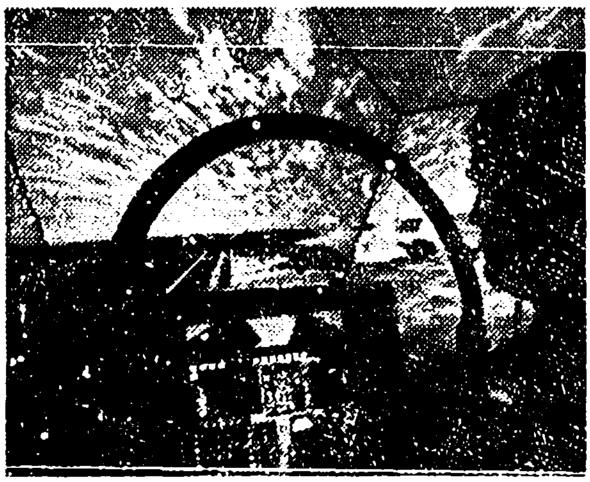


Figure 2. McDonnell Douglas Reconfigurable Cockpit (MDRC) installed in Display for Advanced Research and Training (DART).

The MDRC has a high fidelity F-15 software suite that is derived directly from the McDonnell Douglas engineering simulators in St. Louis. The software includes an F-15 aerodynamics package, a full assortment of air-to-air weapons, a complete radar package, a radar warning receiver (RWR), a HUD, electronic counter measures (ECM), and electronic counter-counter measures capabilities (ECCM). The MDRC provides high fidelity simulation only for air combat functions: there are no rudder pedals or provisions for landing, refueling, or emergency procedures.

F-15 Visuals

Computer Image Generation imagery was provided by the General Electric Advanced Visual Technology System (AVTS) which was the engineering prototype for the CompuScene 4. AVTS provides 8000 faces distributed among ten channels at 60 Hz.

Displays - One MDRC was installed in the McDonnell Douglas full field-of-view dome. This system consists of a 24' diameter dome with 360° horizontal by 190° vertical coverage. The display system incorporates six background projectors with a resolution of 4.3 arc-min/pixel and a 40°, head tracked area of interest (AOI). Resolution in the AOI was 2.4 arc-min/pixel. Only tne AOI and the three forward channels were used during the TRUE resulting in an 210° horizontal by 100° background field-of-view. The AOI could be slewed throughout the dome. The other MDRC was installed in the Armstrong Laboratory display for advanced research and training (DART). The DART is a dome-like display system consisting of eight segments of a dodecahedron which surround the cockpit (Thomas, Reining, and Kelly, 1992). Each segment is a rear projection screen approximately 1 meter from the pilot's head (see Figure 2). During the TRUE, imagery was projected onto only six of the screens at a time as controlled by a head tracker. The DART's field of view during the TRUE was 300° horizontal by 200° vertical with resolution of 4.75 arc-min/pixel. Unlike the dome pilot, the DART pilot could not see to the rear of the aircraft.

Opposing Forces Cockpits

Two Armstrong Laboratory combat engagement trainers (CETs) were integrated into the network as opposing airborne interceptors. CETs are limited function F-16 trainers equipped with air-to-air radar, AIM-9 missiles, and radar warning receivers (Boyle and Edwards, 1992). For the TRUE, the F-16 aerodynamics simulation was replaced with aerodynamic and engine characteristics of an Su-27 interceptor.

Air Weapons Controller Station

The Simulated Command and Control Environment Networked Training System (SCCENTS) was developed by the Logistics Research Division of Armstrong Laboratory to provide the Air Force with a low-cost command and control workstation for research, development, and training. The system was designed to be integrated to networks such as SIMNET or Distributed Interactive Simulation (DIS). The SCCENTS has two display modes - an Airborne Warning and Control System (AWACS) or the 407L Ground Control Intercept (GCI) display. The AWACS display was used for the TRUE study because it provided more physical information about the battle to the controllers. The SCCENTS also has a digital database developed from Defense Mapping Agency data.

Computer Generated Threats

The F-15 and opposing forces cockpits, visual image generators, displays, and the controller station were existing systems which were adapted for network operation. The automated threat engagement system (ATES) was developed especially for the MultiRAD effort (Rogers, 1992). The ATES simulates ground and air threats plus friendly aircraft. Ground threats include: headquarters functions with early warning radar, directed and autonomous surface-to-air missile (SAM) batteries (SA-4, SA-6, and SA-8) with their radars, and ZSU-23 anti-aircraft artillery (AAA). ATES air threats used in the TRUE were Su-27 interceptors with radar and infrared guided missiles and MiG-27 attack fighters equipped with radar jammers. ATES also supplied four F-16 strikers used during offensive counter-air missions. While the ATES hardware was specifically developed for MultiRAD, the threat models and integrating software are "a blend of several programs from various government agencies and a commercial vendor," (Rogers, 1992, p.303).

Exercise Control

Multiship simulation exercises were directed from a central console which contained systems to set up, initiate, observe, videotape, and terminate sorties. Initial conditions and actions of automated opposing forces were preprogrammed in the ATES. Initial conditions for all manned players are programmed in the exercise control station. Monitors at the control station displayed: 1) an overhead view of the engagement, 2) each F-15's front panel including radar, radar warning, and armament control displays, and 3) one channel of out-the-window video (the AOI for the dome and the forward channel for the DART). The test director had intercom communication with each player station. Incorporated into the exercise control station was a computer controlled, video taping system which recorded the two F-15 front panels and the overhead view plus all radio communication. Computer control allowed synchronized start, stop, and playback of the three videotapes.

Debriefing station

After each simulator session, the two F-15 pilots and the air weapons controller would take the videotapes to an independent debriefing room which contained three computer controlled tape players and monitors. After brief instruction, participants were able to review their engagements and could stop, rewind, and replay segments of particular interest. Since the debriefing system was independent of the simulators, a second team could fly while the first was debriefing.

SYSTEM DEVELOPMENT AND INTEGRATION

Network Interface Units

The NIU provides a method of communicating between the host simulator and the MultiRAD network (SIMNET). The NIU and host communicate with one another according to a predefined *language* which is described in the system's Interface Control Document (ICD). The NIU uses ethernet as the physical medium to communicate both with the host computer and SIMNET although other mediums such as Fiber Distributed Data Interface (FDDI) and reflective memory are available for use. The NIU's primary functions include coordinate system conversion, Remote Vehicle Approximation (RVA), data filtering, and conversion of units of measure.

Armstrong Laboratory developed one NIU for each of its networked simulation systems. The NIU's hardware consists of two Motorola 68030 single board computers with transition modules, an enhanced ethernet interface board from CMC Corporation, a SIMVAD digital voice processing board, a SIMVAD voice communications adapter with a headset, a twelve-slot VME chassis, and a removable disk drive assembly. One computer board processed simulation data between the SIMNET network and the simulation host while the other board processed digital voice data between the host and the network. The CMC ethernet processor board was chosen for communications between the NIU and SIMNET because it was faster than the MVME712 transition module interface. The CMC boards also contained firmware to monitor ethernet statistics such as collisions and bad packets.

Communications Protocol and Extensions

SIMNET was developed by the Army and ARPA for tanks and slow moving air vehicles. Armstrong Laboratory implemented a subset of the SIMNET Protocol Data Units (PDUs) specific to air combat operations. Initially, the activate request and response, deactivate request and response, vehicle appearance, fire, and impact PDUs were implemented and tested. After all systems were able to communicate with each other and observe each other in the synthetic battlefield, the protocols were extended to include freeze, radar, and emitter PDUs. The MDRCs provided the most fidelity and were chosen to provide the testbed for implementing the protocol extensions. The MDRCs were first tested one on one, then integrated to the other devices on the network.

All SIMNET PDUs were modified to add a time stamp field. Additionally, fifteen extra result types were added to the impact PDU to enhance scoring capabilities. Originally, the SIMNET protocol only had four results for scoring: miss, ground impact, vehicle impact, and proximity impact. A freeze PDU was also implemented to allow the exercise control station to stop (freeze), and continue any mission.

The radar and emitter PDUs were designed to pass all the information needed about a specific radar or emitter over the network to other vehicles. The radar PDU includes radar system, radar mode, radar ID, sweep, power, and a list of illuminated targets. The emitter PDU includes the number of emitters, emitter class, mode, power, frequency, and sweep. Air-to-air tacan, IFF, jammer, and radio emitters were specifically implemented on the network. One issue that had to be addressed was the classified nature of the radars and emitters of the MDRCs. Should classified data be sent over the network or do all players have classified information about every other player on the network? Armstrong Laboratory implemented the PDUs such that individual packets were not classified but aggregates of packets may be classified.

Each system had different levels of computing ability. The MDRCs originally could only handle eight threat vehicles, eight missiles, and eight ECM bodies. After upgrading the system to Motorola 88100 processors, the system was increased to handle fifteen threats, eight network missiles, eight internal missiles, eight network ECM bodies, eight internal ECM bodies, and eight SAM/AAA sites. The CETs on the other hand

could only handle six network vehicles. A priority routine was used to monitor the closest threats. Similar limitations existed in the visual systems where priority schemes were also implemented.

While integrating the MDRC to the network, a single vehicle update message was designed to transfer simulation information about all entities on the network. This single vehicle update message contained position, velocity, and state information including radars and emitters. However, ECM and missile entities do not need to pass information such as radars, emitters, and throttle position. A second, streamlined, vehicle appearance message was created for ECM and missiles to reduce the amount of traffic between the host and NIU.

Problems Encountered

Many unexpected problems arose during the integration of the systems for the TRUE. For example, the CETs and ATES were accepting any impact message containing their vehicle ID as a kill without looking at the result. Under certain circumstances, the MDRCs would kill vehicles when the gun trigger was depressed- even though the vehicle was 80 miles away. Ghost vehicles were also created on the network due to improper memory management. Still another memory management problem allowed vehicles to attain attributes of vehicles that were previously in that portion of memory. A CET might fly a jamming mission in one sortie; in the next sortie, whichever vehicle occupied the same memory space had the jammer flag set.

During the TRUE, networked vehicles jittered in the visual systems. Several reasons for jitter were identified. One portion of the jitter occurred when a vehicle exceeded the RVA dimensions. Jitter was also caused by the different frequency rates of the devices on an asynchronous network. Still another cause for jitter was simulator or NIU overloading. When a device could not send and receive packets at its predefined update rate, it would take large jumps in the visual systems. Finally, coordinate conversions and precision were found to be contributors to jitter. In the early stages of the TRUE, some vehicles were making large jumps in the visuals. The coordinate transformation algorithms in the NIUs were found to be incorrect. All systems were analyzed and modified to reduce the possibility of overloading. The scenarios were also analyzed to ensure devices did not overload due to network traffic. Additionally, the coordinate transformations were corrected. The resulting jitter was deemed acceptable to the program because it was minor. A smoothing algorithm could be used to further reduce the jitter but was not implemented due to system loading.

TEST AND EVALUATION

The objective of the TRUE was to evaluate the strengths and weaknesses of the MultiRAD system. The primary data for this evaluation were pilot comments from daily debriefings and questionnaires completed after each simulator session. These comments were used to improve the MultiRAD system after each week of the TRUE. Training effectiveness was evaluated by relating pilot and air weapons controller ratings of system utility to rated training effectiveness. These evaluations would be used to determine whether the training benefits seen in the McDonnell-Douglas Advanced Air Combat Training exercises could be obtained using the MultiRAD system.

Procedures

The TRUE consisted of four, one-week training exercises for teams of F-15 pilots and air weapons controllers. The TRUE exercises were conducted in Oct., Nov., and Dec. 92 and Jan. 93. Three or four teams participated each week, with a team consisting of a lead pilot, a wing pilot, and a controller. Each team flew offensive and defensive counter-air missions against a force of up to six aircraft plus surface threats. During each of seven simulator sessions, a team flew their mission three or four times with different tactics used on each setup. After each simulator session, teams reviewed videotapes of the engagements and completed an evaluation questionnaire. Participants were also asked for their evaluation of the MultiRAD system

during daily meetings and during individual interviews.

Participants

Twenty-three, USAF, F-15 pilots and thirteen air weapons controllers participated in the TRUE exercises. Pilot experience levels ranged from 300 to 2500 total flying hours with a median of 1400 total hours and 675 F-15 hours.

RESULTS

During the four weeks of the TRUE, 78 hours of multiship simulator exercises were scheduled and 72 were completed; six hours (8%) were lost to major systems failures. Within the 72 hours, 267 multiship setups were conducted. Of these, 204 setups were completed successfully while 63 (24%) required a restart due to minor system failure. The proportion of setups which experienced minor failures dropped from 30% during the first week to 21% during the other three weeks. The results of the TRUE exercises are described in terms of training utility by Berger and Crane (1993) and comparisons of the DART and dome visual display systems by and Crane (1993). The focus of this paper is on network performance, utility of system components, and modifications to the system in response to pilot comments.

Network Analysis

Network traffic was captured for the final week of the TRUE to determine network loading and characteristics. Table 1 summarizes the data captured for the three engagement types used in the TRUE: basic fighter maneuvers (BFM), defensive counter-air (DCA), and offensive counter-air (OCA). The BFM engagements were one-on-one

Table 1. MultiRAD Network Utilization (Kbits/sec)

	BFM	DCA	OCA
Min	42	45	56
Max	204	294	337
Avg	50	73	87

fights between the two MDRCs. The average utilization for all engagements was less than one percent while the maximum utilization was never more than 3.4 percent. Additionally, the network exceeded 90 percent of the maximum values only two to three times per engagement. The network analysis done by the Laboratory showed no collisions, no cyclic redundancy check errors, and no alignment errors. This was probably due to the fact the network was not heavily loaded.

The largest number of PDUs on the network was the vehicle appearance PDU followed by the voice, radar, and emitter PDUs (See Table 2). The deactivate, fire, and impact PDUs are all event type PDUs and each took less than one packet per second during the TRUE. The network loading is consistent for the three different scenarios. The number of entities on the network is dependent on pilot input. The more ECM and missiles are deployed, the more entities are placed on the network. The average number of entities for each type engagement was 24 entities for each BFM, 38 entities for each DCA, and 39 entities for each OCA.

Table 2 Average Network Packet Utilization (Pkts/sec)

	BFM	DCA	OCA
Vehicle Appearance	30	38	47
Radar	1	4	4
Emitter	0	1	2
Voice	5	16	16

Remote Vehicle Approximation (RVA) algorithms were used to reduce the amount of data on the MultiRAD network. The algorithms use linear interpolation to determine a vehicle's new position. All moving entities on the network had RVA models of 10 meters long, 20 meters wide, 1 meter high, and 3 degrees rotational. If a model's delta between the actual position and RVA position exceeded 10% for any of these dimensions, an appearance packet was sent to all other vehicles on the network. The

RVA algorithms reduced network traffic by 65 to 85 percent over the course of the TRUE. Because tactics and maneuvers are different for each engagement, the effectiveness of RVA on network loading was also different for each engagement.

Although the network was near perfect, a hardware discrepancy in some of the NIU's ethernet boards was discovered early in the program that affected network reliability. The descrepancy was a result of the fact that the only contacts between the circuit board and the connector were solder joints on the pins. After repeated use of the connectors, the solder joints would fail, resulting in loss of data. This loss of data caused unpredictable operations on the network and large amounts of jitter. Once the connectors were repaired by the manufacturer, the problem was resolved.

Component Utility

F-15 Cockpits - The MDRC cockpits used by the TRUE pilots were rated as wholly acceptable for air combat training. The glass cockpit and touch panel were downrated only in that the displays were not positioned exactly as in the aircraft, and pilots had to scan for a moment to find what they were looking for. The lack of rudder pedals was cited as a problem only in close combat which was not a MultiRAD objective. A major difficulty, however, was that some of the avionics software in the simulator was not current with the aircraft. Pilots complained vigorously that the older software prevented them from using their weapons systems as they would in the aircraft. This lack of currency affected the pilots' tactics and greatly reduced the value of MultiRAD training.

Visual Display Systems - Pilots rated a wide field-of-view visual display system as necessary for effective multiship training even when the training objectives stressed beyond visual range tasks. Wide field-of-view visuals are necessary to maintain tactical formation, in transition from medium to short range weapons, to maintain mutual support, to disengage and reattack air targets, and to defend against surface-to-air missiles. Neither the DART nor dome were fully acceptable; however, the DART was preferred. While the AOI in the dome has higher resolution than the DART, the low resolution dome background would prevent a pilot from seeing his wingman without turning his head and searching with the AOI. Also, the DART's higher brightness and contrast increased pilot acceptance. F-15 or Su-27 size aircraft are reduced to an image of one or two pixels at 0.5 - 1 nautical miles. To increase the range at which aircraft could be detected and identified, a simulator unique effect (i. e., a simism) was introduced. At maximum visual detection range, an air target was represented as a white point light. At maximum identification range, enemy aircraft were represented as red point lights while friendlies were represented as blue, flashing lights. This simism greatly increased pilot acceptance; however, pilots continued to complain that they could not determine another aircraft's range or aspect until it was within 0.5 - 1 nautical mile.

Opposing Forces - Houck et al. (1989) found in the Advanced Air Combat Training program that the most significant training benefits came from the opportunity to engage multiple bogeys. However, in the first week of the TRUE, participants found so many flaws in the representation of opposing aircraft that they rated the training to have little or no value. TRUE pilots stated that both the manned and computer generated threat pilots had perfect situation awareness, flew aircraft that were too fast, had 360° radars which could not be defeated, and fired invincible missiles. Many of these problems were found to originate from aircraft and radar models used in the ATES which were generated from unclassified and widely available reference sources. While each parameter used in these models may have been only slightly optimistic, combining them all into a single threat model produced an unbeatable foe. Adjusting the threat model parameters using better data or pilot acceptance greatly increased MultiRAD training effectiveness ratings during the remainder of the TRUE.

One difficulty which could not be corrected was the infrared guided missile model used by the manned opposing forces

pilots flying CETs. This model was originally developed for intercept training against a non-responding target. If the CET pilot has correctly positioned his aircraft with respect to the target, the missile scores a kill. When integrated into MultiRAD, however, the target aircraft's pilot would attempt to defeat the missile by effecting counter measures. These counter measures had no effect on the CET's missile. Pilots greatly objected to this aspect of MultiRAD simulation since it prevented them from practicing their skills at missile defense. The only corrective measure possible during the TRUE evaluation was to brief the CET pilots to hold their shots until the probability of a kill was very high. This solution satisfied no one.

Air Weapons Controller Station - Although the SCCENTS station provides only a functional representation of an AWACS or GCI display rather than a high fidelity physical simulation, the controllers who participated in the TRUE gave uniformly high ratings to the training provided by MultiRAD. During interviews, controllers stated that the SCCENTS station did present a number of simisms to which they quickly adapted. The most significant simism was that the simulated radar was too good. Aircraft altitudes were identified too precisely and there were no blind areas behind terrain features. The training value in MultiRAD came from the opportunity to practice several setups within a simulator period and then to debrief these engagements with the pilots while watching the video tapes and listening to their radio calls.

Debriefing Station - Pilots and controllers rated the computer system for controlling the synchronized videotapes as being overly complex. Only a few participants actually learned to use the system to its full advantage. Re-designing the system to make it fighter pilot friendly is a high priority item for improving MultiRAD. Aside from the complexity, however, pilots agreed with the controllers that reviewing video tapes of the engagement added greatly to the value of MultiRAD training. In particular, seeing the overhead view together with their own radar allowed pilots to evaluate the effectiveness of their actions in context of the entire mission.

LESSONS LEARNED

Functional vs. Physical Fidelity - Boyle and Edwards (1992) point out that, "A real program killer is unmet pilot expectations. If you are not simulating it, don't try to make it look like you are," (p. 496). This lesson was repeated during the TRUE in that functions which were incompletely modeled but were central to the missions being practiced raised the most serious objections. Notably, the lack of currency between the pilot's aircraft and the simulated F-15 software raised howls of protest. Pilots were compelled by the simulation to practice using tactics that they would not use if they were to go to war tomorrow. In this case, reduced fidelity was not good enough. Compare this response to the glass cockpit MDRC. The front panel was a CRT there were no rudder pedals, and most switches were missing. None of these faults affected the mission and they caused no objections. Systems which were critical to the mission, hands-on-throttle-and-stick (HOTAS) controls and the displays used in air combat, were high fidelity simulations. The net effect was a fully acceptable trainer, except for the software. Air weapons controllers using the SCCENTS station had a similar experience. The physical similarity between the SCCENTS and an actual AWACS or GCI station was limited to the mission critical information on the display. Controllers adapted to the simisms caused by the non-critical elements of the SCCENTS and rated the system as providing high value training.

Integration of existing systems - The CET used as a manned opponent in the TRUE is a part task trainer designed to teach the shooter to intercept an air target and to put his aircraft into an optimal firing position. The objective of this part task trainer is to teach the pilot to make the intercept and to take a good shot. The missile model provides feedback to the pilot about his intercept: a good shot gets a kill and a bad shot misses. This missile model completely fulfills the CET's training objectives. The objectives of MultiRAD training, however, include training the F-15 pilot to defend against air-to-air missiles. The IR missile models used by the ATES were sensitive to flares and the target aircraft's throttle setting so that the F-15 pilot

could, if he was quick enough, defeat an ATES missile. CET missiles were not designed to provide defensive training for the target aircraft's pilot and were therefore not responsive to flares or other counter measures. The CET's missile is not a low fidelity model within its intended application. Integrated into MultiRAD, however, the CET missile was unacceptable. Integrating existing systems requires very careful consideration of each system's original objectives and how that system operates. Networking existing systems will support effective training only if the objectives of the integrated system are clearly stated and the capabilities of the individual components are evaluated in terms of these objectives.

MultiRAD Network Limitations - The simulation network was not a limiting factor on the current MultiRAD system. Network utilization for the TRUE study never exceeded 3.4 percent of the full capacity of ethernet. With bandwidths of 100 Megabits/sec available for FDDI, the network should continue to fulfill the requirements of mission rehearsal and team training. The limiting factors found during the TRUE study were: host to NIU interface, NIU processing capability, and host computer processing capability. The MDRC ICDs were developed to maximize the amount of valuable information passed to and from the network. As the number of entities on the network increases, the simulator and NIU spend more time transferring and processing this data. Data minimization methods must be developed to optimize not only the data that is sent to and from the host but also the frequency of these transfers. Also, the network analysis showed the NIU spent the largest amount of time building up the host data buffer. For example, it took an average 49.5 percent (24.75 msec) of the total frame for the NIU to build the MDRC's data buffers. The primary restriction encountered in setting up the TRUE scenarios, however, was the physical limitations of the host simulators' computing power.

Test procedures and Configuration Control - As the integration process progressed, it became apparent that more testing and configuration control were needed. The time between the TRUE scenarios was not always adequate to thoroughly test changes to the overall system. Additionally, the integration effort was accomplished by four separate contractors and the government. Some problems occurred due to miscommunications between the integration teams while others were a result of misinterpretations of the protocols and ICDs. Still others were due to poor configuration control of software. Problems encountered and fixed would sometimes reappear in scenarios Thorough test procedures and rigorous configuration control are crucial for an integration effort of this type.

LIST OF ACRONYMS

AAA--Anti-aircraft artillery
AOI--Area of interest
ARPA--Advanced Research Project Agency
ATES--Automated threat engagement system
AVTS--Advanced visual technology system
AWACS--Airborne warning and control system
BFM--Basic fighter maneuvers
CET--Combat engagement trainer
DART--Display for advanced research and training
DCA--Defensive counter-air
DIS--Distributed interactive simulation
ECM--Electronic counter-measures
ECCM--Electronic counter-counter-measures
FDDI--Fiber distributed data interface
GCI--Ground control intercept
HOTAS--Hands-on-throttle-and-stick
HUD--Head-up-display
ICD--Interface control document
IFF--Identify friend or foe
IR--Infra-red
MDRC--McDonnell Douglas reconfigurable cockpit
MultiRAD--Multiship research and development program
NIU--Network interface unit
OCA--Offensive counter-air
PDU--Protocol data unit
RVA--Remote vehicle approximation
RWR--Radar warning receiver
SAM--Surface-to-air-missile
SCCENTS--Simulated command control environment networked training system
SIMNET--Simulation Network

SIMVAD--Simulation voice analog digital
TRUE--Training requirements utility evaluation

REFERENCES

Berger, S. and Crane, P. (1993). "Multiplayer Simulator Based Training for Air Combat." Proceedings of the 15th Interservice/Industry Training Systems and Education Conference. Orlando, FL: National Security Industrial Association.

Boyle, G. and Edwards, B. (1992). "Low Cost Trainers: Lessons for the Future." Proceedings of the 14th Interservice/Industry Training Systems and Education Conference. San Antonio, TX: National Security Industrial Association.

Crane, P. (1993). "Evaluation of Two Wide Field-of-View Display Systems for Air Combat Training." Presented at: Euro Display '93 - 13th International Display Conference, Strasbourg, France.

Houck, M., Thomas, G., & Bell, H. (1989). "The training potential of multiplayer air combat simulation." Proceedings of the Human Factors Society 33rd Annual Meeting. Santa Monica, CA: The Human Factors Society.

Rogers, B. (1992). "Tactical Air Threat System for a Distributed Simulation Network." Proceedings of the 14th Interservice/Industry Training Systems and Education Conference. San Antonio, TX: National Security Industrial Association.

Thomas, M., Reining, G., & Kelly, G. (1992). "The display for advanced research and training: An 'inexpensive' answer to tactical simulation." Presented at: International Training Equipment Conference, Birmingham, England.

INTERDEPENDENCE OF TRAINING UTILITY AND NETWORK PERFORMANCE USING THE ARMSTRONG LABORATORY MULTISHIP RESEARCH AND DEVELOPMENT SYSTEM

Mr. Thomas L. Gehl, IBM Federal Systems Company; Captain Richard L. Rogers, AL/HRAE; Captain Mark A. Miller, AL/HRAD; Mr. Joseph Rakolta, Loral Defense Systems - Akron

ABSTRACT

To determine the value of a training system we must evaluate the system's design and performance with respect to the training effectiveness needed to support the operational mission. We will need a means to determine the relationship between a system's engineering design parameters and the training utility during a specified mission scenario. Through the research efforts of Armstrong Laboratory's Aircrew Training Research Division, we will address this need by using a networked multiship simulation system with experienced mission ready pilots, including Desert Storm veterans, flying specified mission scenarios. We will then relate network performance measurements to the evaluation of the training utility for critical segments of the mission scenarios.

We will also discuss the relationship between the training utility and network performance for specified mission scenarios. We will characterize the architectural components of the Multiship Research and Development (MultiRaD) training system and define the mission scenarios developed for the MultiRaD training utility evaluation. We will describe the test cases for measuring the network performance and present results of the network performance results with both average and worst-case segments of the mission scenarios. Finally, we will evaluate the network performance results with respect to the training utility and will recommend methods of extrapolating the results to future systems.

BIOGRAPHIES

Thomas L. Gehl is a lead system engineer for IBM Federal Systems Company Manassas research and development effort in system integration and networking core competencies. He is responsible for integrating and demonstrating real-time networked applications such as distributed interactive simulation. Mr. Gehl received his BS in Electrical Engineering and his MS in System Engineering from Virginia Polytechnic Institute and State University

Captain Richard L. Rogers is Chief, Engineering Support Section, Aircrew Training Research Division, Armstrong Laboratory, Williams AFB, Arizona. He is responsible for the engineering support of advanced aircrew training systems and simulation research and development. Captain Rogers received a BS in Computer Science from the US Air Force Academy and a MS in Computer Engineering from the Air Force Institute of Technology.

Captain Mark A. Miller is Assistant Chief, Technology Development Branch, Aircrew Training Research Division, Armstrong Laboratory, Williams AFB, Arizona. He is responsible for managing the division's Multiship Research and Development program. Captain Miller received a BS in Engineering from the University of California, Los Angeles.

Mr. Joseph Rakolta is an Engineer with Loral Defense Systems - Akron, currently working at the Aircrew Training Research Division, Armstrong Laboratory, Williams, AFB, Arizona. His responsibilities include development and integration of networking hardware and software in support of the MultiRaD program. Mr. Rakolta holds a BS in Aerospace Engineering from the University of Cincinnati.

INTERDEPENDENCE OF TRAINING UTILITY AND NETWORK PERFORMANCE USING THE ARMSTRONG LABORATORY MULTISHIP RESEARCH AND DEVELOPMENT SYSTEM

Mr. Thomas L. Gehl, IBM Federal Systems Company; Captain Richard L. Rogers, AL/HRAE; Captain Mark A. Miller, AL/HRAD; Mr. Joseph Rakolta, Loral Defense Systems - Akron

RELATIONSHIP BETWEEN TRAINING UTILITY AND NETWORK PERFORMANCE

For this research effort, Armstrong Laboratory implemented a low-cost multiship simulation system as a training utility for offensive and defensive counterair mission scenarios. The network performance measurements occurred while teams of mission ready pilots flew specified mission scenarios in a realistic combat environment for a training utility evaluation.

As the missions were being performed, we measured the performance of the network system which provided the ability for the teams to play together in a simulated combat environment. To ensure unbiased results, we measured the network performance, independently while the pilots flew their training sorties. The Training Requirements Utility Evaluation (TRUE), which was the training utility that the performance measurements were based on, was designed to determine the training potential of the MultiRaD system. Thus, we measured the network performance of the mission scenarios used to evaluate the training effectiveness of the system.

TRAINING SYSTEM ARCHITECTURE

Armstrong Laboratory designed the MultiShip Research and Development (MultiRaD) system to provide an environment that supports research into the effectiveness of using a network of simulators to provide aircrew training. The MultiRaD system integrates aircrew training simulators and an automated threat simulator over an asynchronous Ethernet Local Area Network (LAN) on which the SIMNET version 6.6.1 protocol with extensions communicates. The system consists of four aircraft simulators and their associated visual systems, a Ground Control Intercept (GCI), an Exercise Control Station (ECS), and an Automated Threat Engagement Simulator (ATES) interconnected over the Ethernet LAN as shown in Figure 1. For a more detailed description on the MultiRaD system see the paper titled "Development, Test and Evaluation of a Multiship Simulation System for Air Combat Training" by Captain Philip Platt and Dr. Peter Crane.

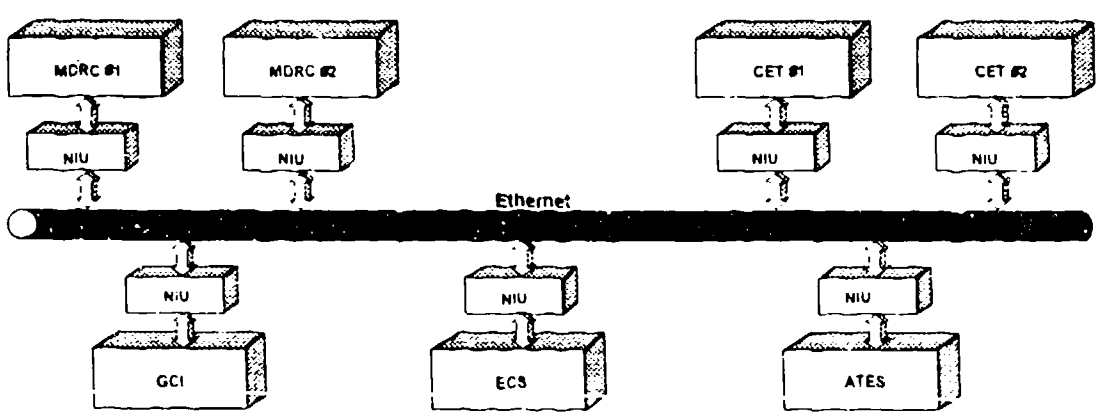


Figure 1. MultiRad System

Simulators

The manned flight simulators consist of two McDonnell Douglas Reconfigurable Cockpits (MDRCs) configured as F-15Cs and two Combat Engagement Trainers (CETs) configured as F-16Cs but visually represented on the network as SU-27s. The station simulates an AWACS controller station and provides intercept vectoring to the F-15Cs. The ATES provides various surface-to-air missile (SAM) threats as well as autonomous intelligent flight models (IFMs). The ECS collects and reproduces exercise data and provides an overview display of the gaming area to allow real-time control of the exercise. In the TRUE, Blue force consists of the two manned MDRCs and ATES provided Blue IFMs, and the Red force is made up of the two manned CETs and ATES provided surface-to-air missiles (SAMs) and Red IFMs.

Visual Systems

The visual systems for the aircraft simulators consist of a 24 ft. Full Field of View (FFOV) dome, a Display for Advanced Research and Technology (DART), a Mini-DART and a CRT. During this study, one MDRC used the FFOV dome which displayed four visual channels using a high resolution Area of Interest (AOI) headtracked over the complete dome and inset into lower resolution front, left and right channels. The other MDRC used the DART which displayed six visual channels on eight display screens, switching imagery from two of the front screens to two of the rear screens using headtracking. The CETs used the Mini DART, using only the front screen, and the CRT which provided only single channel displays. Even though the visual systems varied for each of the manned simulators, they were not evaluated in our analysis as to their effects on network or system performance. For an evaluation of the visual system effectiveness, see the paper titled "Visual Training Requirements for Networked Fighter Simulators, using Armstrong Laboratory's F-15 Training Requirements Utility Evaluation", by Captain Mark Miller.

Network Architecture

Each simulator communicates over the Ethernet LAN through a Network Interface Unit (NIU) which implements the SIMNET 6.6.1 protocol with Armstrong Laboratory extensions for air-to-air combat (i.e. SIMNET 6.6.1+). SIMNET 6.6.1+ defines the transport functions and the application information for the simulators to communicate. The transport functions are provided by the SIMNET Association protocol, and the application information is provided by the SIMNET Simulation protocol data units (PDUs) with extensions for RADAR and Emitter PDUs to support air-to-air combat. The MultiRaD simulators utilize the Activate Request, Activate Response, Deactivate Request, Vehicle Appearance, Fire and Impact PDUs from SIMNET 6.6.1 and the Radar, Emitter, and Freeze PDUs defined by Armstrong Laboratory to intitiate, control and communicate state information of the simulated world. In comparing the SIMNET PDUs to the Distributed Interactive Simulation (DIS) PDUs, the Vehicle Appearance is analogous to Entity State, and the Voice is analogous to the Signal PDU.

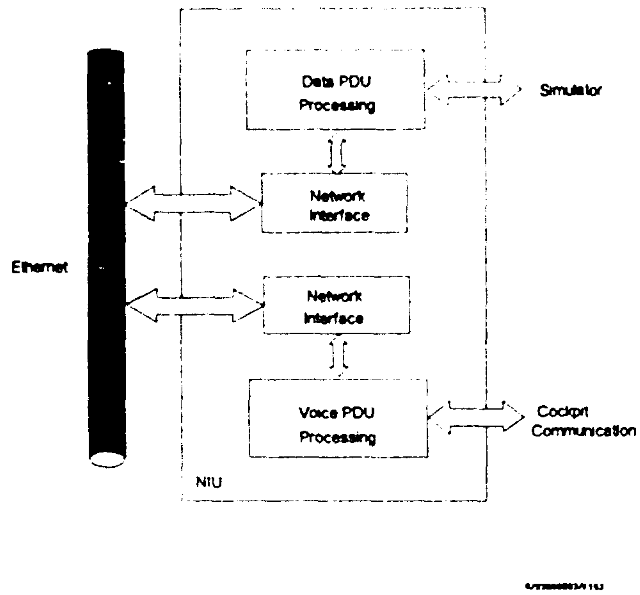


Figure 2. Network Interface Unit (NIU) Function Diagram

Implementing SIMNET 6.6.1+, the MultiRaD NIUs contain two Versa Module Europa (VME) 147 Central Processing Units (CPUs), one Ethernet card, one Simulated Voice, analog to digital (SIMVAD) card, and two VME 712 cards with a functional layout as shown in Figure 2. The two NIU CPUs run separate processes. The first 147 CPU interprets SIMNET 6.6.1+ network traffic via the Ethernet card, translates simulation information, and sends this information to a particular host simulator at a

specified rate. The second 147 CPU processes and sends voice information between the network and the SIMVAD card. As an interface to the SIMVAD card, the 147 CPU processes and concatenates application information to the packetized voice from the SIMVAD card to create the voice PDUs. In the SIMNET NIU, the data and voice PDUs have separate processing paths for the translation and application-type processing.

In implementing the SIMNET protocols, the NIU translates and communicates simulation information between the distributed simulators. The NIU synchronously communicates to each host at a specified frame rate of 20 Hz for the MDRCs, 30 Hz for the CETs, 20 Hz for the ATES, and .1 Hz for the GCI, and asynchronously communicates the PDUs between the distributed hosts over the Ethernet LAN. To communicate between the distributed hosts on the network, the NIU uses group addresses with the multicast service to separate and filter voice and simulation PDUs on the network adapters to decrease the number of PDUs which must be processed by the NIU CPUs. Within each frame rate, the NIU services the host, services the network interface, performs dead reckoning on each entity, checks the thresholds on the host vehicle and passes information to the host. A more detailed description of NIU performance and functions is provided in the NIU Detailed Design Specification at Armstrong Laboratory.

To reduce the network traffic in communicating the Vehicle Appearance PDUs, we implemented dead reckoning or remote vehicle approximation (RVA) schemes in the NIUs. The dead reckoning algorithms extrapolate, linearly in time, the vehicle position based on the last velocity and time information received in the Vehicle Appearance PDU. To perform dead reckoning, the host calculates actual position along with the dead reckoned position and determines if the difference between the actual and dead reckoned positions exceeds a pre-defined threshold. If a threshold is exceeded, the host NIU communicates a Vehicle Appearance PDU to update the rest of the simulators involved. By communicating positional updates only when a threshold is exceeded, the dead reckoning algorithms significantly reduce the network traffic.

While it reduces the network traffic, the dead reckoning algorithms do not cause any apparent visual degradation. In our mission scenarios, the two MDRC simulators can fly formation with little to no jitter problems. For vehicles portrayed by the MDRC, CET and ATES simulators, the dead reckoning vehicle dimension thresholds corresponded to a length of 10m, a wing span of 20m, a vertical distance of 1m, and a rotation of 3 degrees. These lengths corresponded to a threshold of 10% of the actual vehicle dimensions. These vehicle thresholds in conjunction with the actions taken by the pilots on the particular simulators are to a large extent responsible for the frequency at which the Vehicle Appearance PDUs were communicated onto the network.

MISSION SCENARIOS

The mission scenarios for the TRUE represent those used in the Advanced Air Combat Simulation (AACS) at McDonnell Douglas. We intended to create situations where the two manned F-15Cs would meet aggressive threat aircraft and an integrated air defense system and make tough in-flight combat choices to perform their missions successfully.

The missions took place on the TACWAR data base, which is a Defense Mapping Agency (DMA) representation of western Washington state with an effective area of four degrees longitude by four degrees latitude. The designated Forward Edge of the Battle Area (FEBA) was 47 degrees North Latitude. In both the defensive and offensive counterair mission scenarios, the attacking team was based in the North and had an objective to strike the Chehalis Airfield in the South.

There were seven defensive counter-air (DCA) scenarios, and in all cases, the F-15 Combat Air Patrol started forty miles south of the FEBA, knowing that the threat axis was northerly. The objective was to defend the associated air defense lane, and all "high value" targets (i.e. Chehalis Airfield) from strike aircraft. Red aircraft were briefed as escorting SU-27 "Flanker" interceptors and radar jamming Mig-27 "Flogger" strikers. Additional risks in the mission included the possibility of chemical threat in the theater.

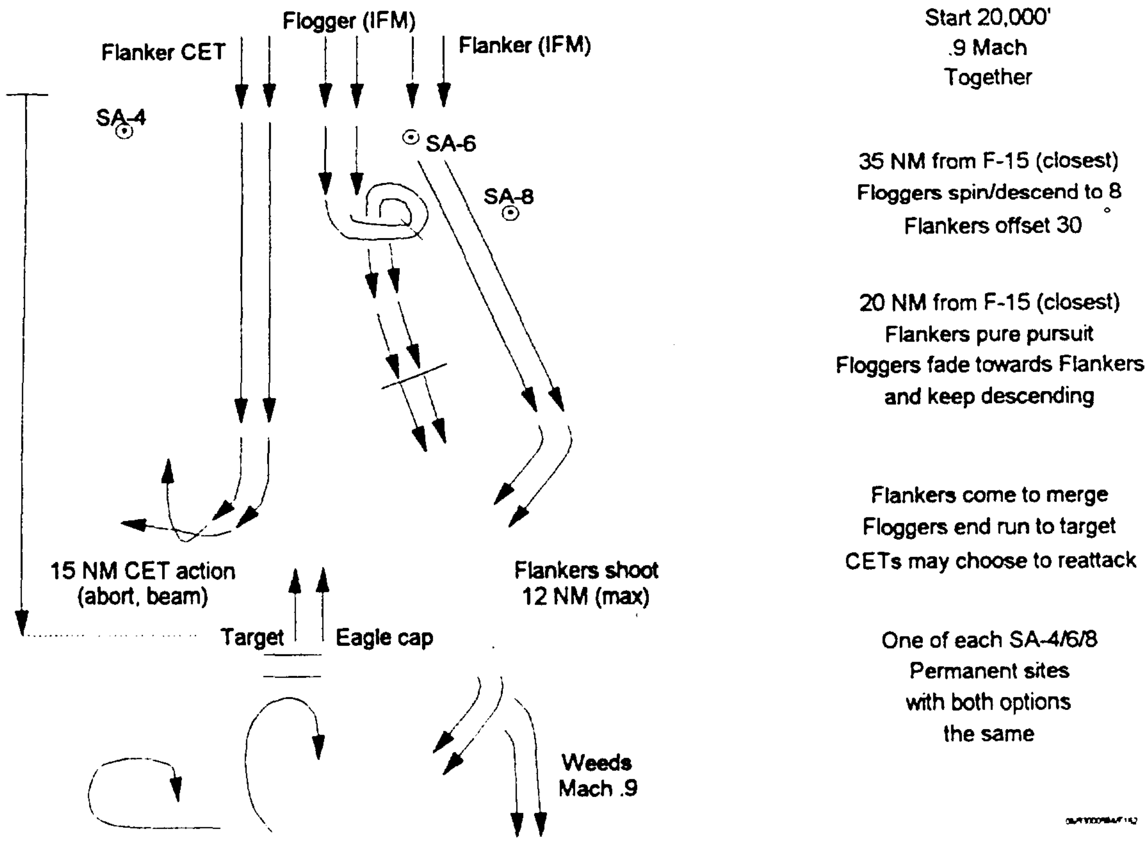


Figure 3. Defensive Counter-Air Champagne Tactic

Threat packages formed forty miles north of the border and ran a coordinated tactic, attempting to get the strikers through to the target. Fixed SAM sites, including the SA-4, SA-6, and SA-8 were situated north of the border within menacing range. Both sides considered a shooting war to be in progress and political borders were not a critical factor. Figure 3 depicts a "Champagne" tactic where Floggers spin to the low altitude block, while the escort attempts to mix it up with defending F-15 Eagle CAP in the south.

Two Flanker threats were provided by the CETs, and the other two Flankers as well as both Flogger aircraft were provided by the ATES. The F-15 pilots were not briefed on which aircraft were manned and which were unmanned or automated. Their challenge was to kill the Floggers north of Chehalis without falling prey to the escorts or SAMs.

In the six offensive counter-air (OCA) scenarios, the F-15s were tasked to escort four F-16s on a high priority bombing mission of Chehalis airfield from which Flanker Interceptors were operating in two-ship formations. The ingress route started 40 miles north of the FEBA and changed direction at the FEBA to head for the airfield. Figure 4 shows three Combat Air Patrols acting in defense of the red homeland. Of the six defenders, two were CETs and four were ATES entities. The four F-16s were also provided by the ATES as were three known SAM sites.

The OCA scenarios defined the maximum number of entities available for the TRUE. Initially, the OCAs required 10 ground threat sites with a mix of Anti-Aircraft Artillery and SAMs. During the integration, this number was dropped to three to enable the ATES to stay

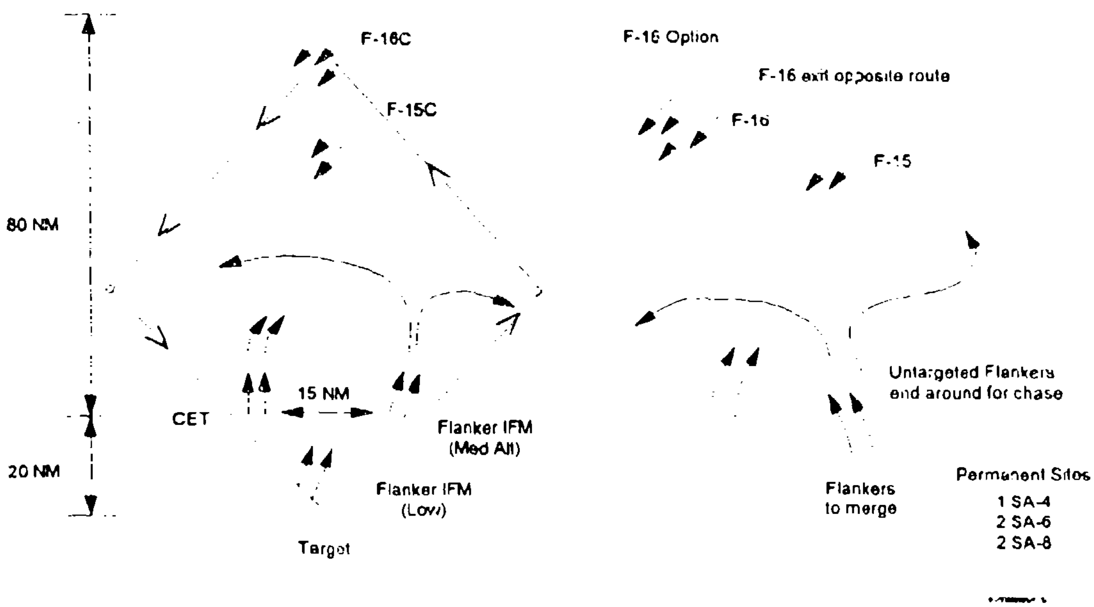


Figure 4. Offensive Counter-Air Maneuver

within the frame cycle of the simulation. In addition to the maneuvering during the engagements, both sides fired multiple missiles and the MDRCs dropped chaff and flares. We ensured that the systems stayed within their simulation frame rate while keeping the scenarios large enough to remain challenging.

TEST CASE AND RESULTS

We designed software tools on a SUN 3/80 and utilized a PC 386 Network Analyzer to measure the network performance during the TRUE studies conducted over several months. With the SUN 3/80, the network analyzer, and the data collection NIU, we captured a large quantity of network data that was available for post processing. For the purposes of determining results for this paper, we analyzed the data obtained during the last week of the TRUE study, when the system was most stable.

On the SUN 3/80 machine, we developed a software program to record network packet source Ethernet address, network packet length, SIMNET 6.6.1 packet type, and time of packet arrival to be used to determine the network performance during each simulation mission scenario. In conjunction, we monitored the network collisions, fragmented packets, misaligned packets, bad cyclic redundancy code (CRC) checks, and lost packets for each scenario, using the network analyzer to evaluate the degradation of the Ethernet performance. Finally, we modified an NIU to acquire NIU internal processing timing data.

PDU Rate and Bandwidth Distribution

For the results of this paper, we extracted information on network packet throughput and network bandwidth analysis representing forty-eight of the mission scenario that were conducted during the last week of the TRUE study. For each scenario, the first plot shows the number of kilobits that are transmitted across the MultiRaD SIMNET 6.6.1 network during each second, and the second plot corresponds to the number of network packets that were communicated across the MultiRaD SIMNET 6.6.1 network per second.

As the number of entities that participate in the mission scenario increases, the mean number of packets per second on the network, and therefore, the mean bandwidth of the network, increases proportionally. Also, the network traffic appears to be relatively constant over time with peak durations. With respect to the mission scenarios, those peak durations of packets communicated occurred during active engagements between the pilots and the entities participating in the mission scenario. These peak durations increase the network loading by approximately 2 to 3 times the normal packet rates of the mission scenario. Thus, by

analyzing the PDU rate and bandwidth distributions over time for various mission scenarios, we are able to conclude that burst traffic produces approximately 2 to 3 times more network traffic than normal network loading for the offensive counter-air and defensive counter-air mission scenarios that were flown.

In evaluating the bandwidth required of the network for these mission scenarios, we measured an average bandwidth utilization of about 1 to 2.5% of the total available bandwidth of the 10 Mbps Ethernet LAN. We did not measure any Ethernet degradation for this low utilization, which is expected. Due to these network bandwidth results, we emphasized the analysis on packet throughput with respect to each device and PDU type.

Averaged PDU Rates

To analyze network performance with respect to individual simulators, we averaged the PDU rates of the mission scenarios for each device and each PDU type on the network. The four manned simulators - 2 F-15Cs (MDRC1&2) and 2 F-16Cs (CET1&2)--along with the ATES contributed practically all of the data PDUs for the respective mission scenarios. They also contributed almost all of the voice PDUs on the network; thus, contributing all of the measurable PDUs and bandwidth on the network for the respective scenarios. We evaluated the PDU types of Appearance, RADAR, Emitter, and Voice, explicitly, and grouped the Activate Request, Activate Response, Deactivate Request, Fire, Impact, and Freeze PDUs as Other PDUs. The Other PDUs contribute an insignificant amount over the mission scenarios. We analyzed the PDU rates of offensive counter-air and defensive counter-air mission scenarios of 2 Blue forces versus 4 Red forces (2 V 4), 2 Blue forces versus 6 Red forces (2 V 6), and 6 Blue forces versus 6 Red forces (6 V 6) with varying maneuvers. In these mission scenarios, the MDRCs participate as manned Blue forces (i.e. F-15Cs), the CETs participate as manned RED forces (i.e. SU 27s), and the ATES provides additional Blue and Red forces as IFMs. We averaged 5 missions for each engagement scenario (i.e. 2 V 4, 2 V 6, 6 V 6), to compare the differences between offensive and defensive counter-air network performance (See Tables 1-4).

The manned vehicles contribute significantly more PDUs per entity represented than the unmanned. For manned Blue forces participating in offensive counter air missions, an average of about 17 PDUs per second are transmitted per entity of which about 75% are due to Vehicle Appearance PDUs and about 15% due to Voice PDUs. The manned Red forces transmit about 9 PDUs per second per entity represented with about 75% due to Vehicle Appearance PDUs and 15% due to Voice PDUs. Thus, the manned vehicles have the same distribution of PDUs types communicated, with the Blue forces communicating a larger total amount than the Red forces. This difference between the Blue and Red forces in the total number of PDUs communicated could be due to the fidelity of the vehicles being represented and the maneuvers that they are able to perform. In contrast, the unmanned simulator, ATES, transmits about 2 PDUs per second per entity represented with about 90% due to Vehicle Appearance PDUs and an insignificant percentage due to Voice PDUs (i.e. 1%). This significant reduction of PDUs per entity transmitted results from unmanned simulators using algorithms to control groups of entities with no tightly coupled human interaction to control the maneuvers of the specific entities while the manned simulators have tightly coupled human interaction per entity being represented.

For defensive counter-air, the manned Blue forces transmit about 13 PDUs per second per entity, with 70% due to Vehicle Appearance and 20% due to Voice PDUs. The manned Red forces transmit about 9 PDUs per second per entity, with 68% due to Vehicle Appearance PDUs and 25% due to Voice PDUs. The unmanned simulators transmit 2.5 PDUs per second per entity, represented, with 92% due to Vehicle Appearance PDUs and 1% due to Voice PDUs. The noticeable difference between the analysis of the offensive counter air and the defensive counter-air is the number of total PDUs per second transmitted by the Blue forces. This difference from 17 to 13 PDUs per second, 25% decrease, between the OCA and DCA for the Blue forces could be due to the difference in maneuvers required to perform the mission. For example, more vehicles (i.e. six Flankers instead of four Flankers), more shots, more chaff and more maneuvering requires more state updates, thus, contributing more PDUs.

PDUs/ ENTITY	TOTAL	APPEAR	% APPEAR	RADAR	EMITTER	OTHER	VOICE	% VOICE
TOTAL	73.6	50.2	68%	4.6	1.6	0.5	16.7	23
MDRC1	18.1	14.1	78%	1.5	0.2	0.1	2.1	12
MDRC2	16.8	11.7	70%	1.5	0.2	0.1	3.3	20
CET1	7.5	4.8	63%	0.8	0.2	0.0	1.8	24
CET2	9.5	8.0	84%	0.5	0.1	0.0	0.9	9
ATES	13.0	11.6	89%	0.3	0.9	0.1	0.1	1

Table 1: 6 V 6 (all offensive counter air) - standard deviation of total PDU rate=5.3

PDUs/ ENTITY	TOTAL	APPEAR	% APPEAR	RADAR	EMITTER	OTHER	VOICE	% VOICE
TOTAL	73.6	50.7	69%	5.2	1.9	0.2	15.6	21
MDRC1	17.1	12.8	75%	1.8	0.2	0.1	2.2	13
MDRC2	17.1	12.7	74%	1.4	0.2	0.1	2.7	16
CET1	4.9	2.9	58%	0.8	0.1	0.0	1.2	24
CET2	9.9	8.2	82%	0.9	0.2	0.0	0.6	7
ATES	15.9	14.1	89%	0.4	1.2	0.0	0.1	0

Table 2: 6 V 6 (all offensive counter air) - standard deviation of total PDU rate=2.7

PDUs/ ENTITY	TOTAL	APPEAR	% APPEAR	RADAR	EMITTER	OTHER	VOICE	% VOICE
TOTAL	42.2	24.9	59%	3.1	1.1	0.1	13.3	31%
MDRC1	13.1	8.5	65%	1.2	0.2	0.1	3.1	24%
MDRC2	10.7	6.9	64%	1.3	0.2	0.0	2.3	21%
CET1	2.2	1.7	76%	0.4	0.1	0.0	0.1	5%
CET2	0.1	0.1	100%	0.0	0.0	0.0	0.0	0%
ATES	8.5	7.7	91%	0.1	0.6	0.0	0.1	1%

Table 3: 2 V 4 (2 offensive and 3 defensive counter air) - standard deviation of total PDU rate=1.2

PDUs/ ENTITY	TOTAL	APPEAR	% APPEAR	RADAR	EMITTER	OTHER	VOICE	% VOICE
TOTAL	60.2	40.0	66%	3.9	1.3	0.3	14.7	24%
MDRC1	10.2	6.6	65%	1.1	0.2	0.0	2.3	22%
MDRC2	14.5	10.8	74%	1.6	0.2	0.1	1.9	13%
CET1	[illegible]	5.2	65%	0.6	0.2	0.0	2.0	25%
CET2	[illegible]	8.8	82%	0.6	0.2	0.0	1.2	11%
ATES	9.3	8.6	92%	0.0	0.6	0.0	0.1	1%

Table 4: 2 V 6 (all defensive counter air) - standard deviation of total PDU rate=2.3

The frequent occurrence of Appearance PDUs with an immeasurable number of Fire or Impact PDUs (i.e. event PDUs), emphasize that most of the PDUs communicated are due to the positional updates through the dead reckoning algorithms and not to voice and event occurrences. This conclusion can also be applied to the DIS PDUs since the Vehicle Appearance PDU is directly analogous to the Entity State PDU in the DIS standard.

For the mission scenarios performed, the total PDUs communicated over the network are distributed as 66% due to Vehicle Appearance, 24% due to Voice PDUs, 6% due to Radar PDUs, and 2% due to Emitter PDUs with the rest of the PDUs communicated very infrequently. The difference in the percentages of the total versus the individual simulators is due to the fact that the GCI and ECS transmit voice PDUs in controlling the exercises and the GCI also communicates some Radar PDUs as part of its functions. As the number of entities are increased, the percentage of PDU types will approach the percentages noted for the individual simulators since they will become more of a dominant factor over the quantity of PDUs needed for control functions.

CONCLUSIONS

For this study, we independently measured the network traffic of the TRUE while mission ready pilots flew their specified mission scenarios. Our study demonstrated relationships between the training utility and the network architecture which could be extrapolated for larger operational systems. Due to our preliminary results relative to the training utility, we recommend more direct analysis of pilot performance for specified aspects of the mission scenarios with respect to the network

traffic. These more detailed studies should demonstrate further relationships between the pilot performance, the mission scenarios, and the network traffic.

In summary, we determined that approximately two-thirds to three-quarters of the PDUs communicated are due to Vehicle Appearance PDUs and that practically all of these PDUs were due to positional updates based on the thresholds set for the dead reckoning algorithms. This conclusion can be drawn since an immeasurable number of PDUs were communicated as Fire or Impact which would cause an event update, versus a positional update of the vehicles' appearances.

After analyzing the voice traffic, we concluded that voice PDUs provide 15-25% of the total network traffic when integrated over the same network as the data PDUs. This increase emphasizes that the predominant traffic is due to Vehicle Appearance, and not Voice PDUs, for highly interactive engagements. Also, we found that additional voice traffic can be caused by the controlling of the scenarios but that this additional traffic does not contribute significantly to the overall traffic on the network.

With respect to the training utilities, we found that the capabilities of the aircraft being simulated and the maneuvers required of those aircraft affects the PDU rate transmitted by an individual simulator. The more complex the maneuver causes more non-linear positional updates of the dead reckoning calculations which result in more Vehicle Appearance PDUs communicated to update the most accurate position of ownship. For example, we found that offensive counter air can cause approximately 25% more PDUs to be communicated than defensive counter air maneuvers, due to the increased maneuvering and events (i.e. shots and chaff).

Lastly, as expected, we concluded that unmanned vehicles only require about 2 to 3 PDUs per second to communicate their state changes which is less than manned simulators by a factor of 4 to 8, depending on the maneuvering fidelity performed. This makes sense since the unmanned vehicles are not driven by human interaction, which can be non-linear, and have many vehicles correlated in the algorithms that drive their maneuvers.

To further these conclusions, we recommend additional studies in the analysis of the distributed interactive simulation environment and supporting architectures. One aspect would be to analyze existing architectures in terms of their packet throughput capabilities with respect to the demands of the mission scenario. This study should be extended to gateways and routers, also, to determine the affects of such training over wide area networks. In performing these additional studies, techniques such as multicast and the relationship of group addresses to application information for the varying training utilities could be analyzed to determine the additional reduction of packet throughput.

LOW-COST COCKPIT TRAINER DESIGN: CHALLENGES AND SOLUTIONS

Robert L. Bothwell
James W. Lacy
Lockheed Fort Worth Company
Fort Worth, Texas

ABSTRACT

The challenge to today's training system design engineer is changing; adapting to this change is necessary for survival in a demanding economy. Previously, training system engineers met with success by using emerging technologies to develop ways to increase the capability of training devices: increase fidelity, increase task capacity, increase throughput. The result has been an evolution of larger, more capable, and more expensive training devices. However, in today's environment of declining budgets, another demand is being made of the training system engineer - decrease cost!

The purpose of this paper is to describe specific challenges facing the designer of a cockpit trainer attempting to blend the training requirements of high fidelity and capability with the requirement of low cost. This paper will present innovative methods to overcome these challenges in designing a high fidelity, low-cost, cockpit trainer.

The paper emphasizes the importance of front-end analysis to determine the fidelity and cost factors that would drive the design. Specific examples of training task analysis and preliminary cost determination are given. Specific problems encountered in designing a low-cost cockpit trainer and pragmatic considerations in designing solutions for these problems are addressed. The paper examines alternatives to expensive mechanical instruments and integration and fidelity of virtual displays.

The paper concludes with a discussion of practical benefits of these design solutions. Emphasis is placed on cost savings, reliability, and efficiency through reconfigurability.

ABOUT THE AUTHORS

Robert L. Bothwell is a Senior Engineering Specialist with Lockheed Fort Worth Company. He is a former USAF Instructor Weapon System Officer with ten years experience teaching F-111 operation and tactics in the classroom, in ground-based trainers, and in the air. He has five years experience in developing fighter aircraft trainer requirements and developing and conducting operational tests for aircrew training systems for the USAF. He has four years experience with industry in systems-level design of aircrew training devices. Mr. Bothwell holds a BS degree in Economics from Pennsylvania State University, a BS degree in Computer Science from the University of West Florida, and an MBA from the University of Utah.

James W. Lacy is a Senior Engineering Specialist with Lockheed Fort Worth Company, currently assigned to the staff of the Training Systems group. His current responsibilities are in aircrew trainer technology development including Project Lead on a new low-cost training device concept for the F-16 aircraft, and coordination of group research and development activities. He has over 20-years experience in simulation and aircrew training, beginning with the first Night Carrier Landing Trainer fielded for training. Prior to his four years at Lockheed Fort Worth Company, Mr. Lacy was Project Manager at LTV Aerospace and Defense Company where he managed the Navy's "Top Scene" program, the first real time photo-based mission planning and rehearsal system delivered to the U. S. Navy. Mr. Lacy attended Oklahoma State University and the University of Texas, Arlington, where he obtained a BS degree in Aeronautical Engineering and an MS degree in Mechanical Engineering.

LOW-COST COCKPIT TRAINER DESIGN: CHALLENGES AND SOLUTIONS

Robert L. Bothwell
James W. Lacy
Lockheed Fort Worth Company
Fort Worth, Texas

INTRODUCTION

The traditional relationship between military aircraft training device fidelity and cost has been easy to understand. Historically, as aircrew trainer fidelity and capability have increased, the cost of that trainer has also increased. The cost has been driven primarily by the high cost of computer systems required to support ever increasingly complex aircraft and environmental simulation models. The increasing cost of software development, likewise, increases the price of the models themselves. Training device cost has also historically been driven up by the complexity of the weapons systems themselves. For example, today's front-line fighter aircraft are employing multiple sensors for navigation and weapons delivery. Supporting multiple sensors in a training device can be an expensive proposition calling for multiple displays and correlated data bases. One result of the high cost was the acquisition of trainers without critical major subsystems. The old Tactical Air Command trained its fighter aircrews for years on simulators with no or inadequate Out-the-Window (OTW) visual systems. These devices became little more than avionics procedures or instrument trainers.

In spite of the historical upward trend in the cost of aircrew cockpit trainers, today's military aircraft training system customer is demanding increasing capability in his trainers at lower cost! The increasing sophistication of weapon systems brings with it increasing demands on the training systems. Today's customer wants to train his on-board sensor based systems such as Maverick, GBU-15, and LANTIRN navigation and targeting. He wants his training devices to support training at the edge of the performance envelope for air combat maneuvering and complex tactics. He also wants his cockpit trainers to be interactive to train the synergistic effects and multi ship tactics and related mutual support tasks. On top of all this, the customer would like his training systems to support geo specific environments for mission rehearsal with such features as real-world terrain and cultural features and realistic weather effects.

Training in these sophisticated, edge-of-the-envelope tasks has previously been accomplished by an evolution of large, more expensive training devices. In today's environment of shrinking budgets, the military customer must continue to provide training in his sophisticated weapons in a more cost-effective manner. This need was formalized in a USAF General Officer review of Air Force flight simulator policies on 10 May 1993, which defined an emerging concept of low cost, unit level training devices. It is incumbent upon those of us in the training industry to respond to this emerging need. This paper describes specific challenges that faced the designers of a cockpit flight simulation trainer as they attempted to blend the training requirements of high fidelity and capability with the requirement of low cost. The paper presents methods used to overcome the challenges and the resulting solutions.

DESIGN OBJECTIVES

Our objectives in designing a pilot cockpit trainer for a fighter airplane were straightforward: (1) maximize fidelity and availability; and (2) minimize cost.

Maximize Fidelity and Availability

In our attempts to maximize fidelity, we found we were designing a new class of unit-level trainer. Previous types of unit-level trainers would no longer be acceptable. This design was evolving to be more than a familiarization or procedural trainer focusing on switchology and part-task training. Our design took the approach of integrating high fidelity aircraft systems functionality into a realistic environment.

To achieve accurate aircraft systems performance, we acquired engineering development models used in early design of various systems. To make the trainer fly like the airplane, we used an aero model that had been derived from engineering studies and modified by empirical data from test flights. Weapons ballistics and threat environment models from engineering evaluation facilities were used for ownship weapons flyout calculations and to provide hostile threat environment cues in the cockpit.

With all of these resources available to us, our biggest challenge to meet our high-fidelity requirement turned out to be with the trainer cockpit itself. It would have been incongruous to use these sophisticated models only to present the cues in a lesser fidelity cockpit. In addition, we were aware of user dissatisfaction with less-than-optimum cockpit geometry that compromised training in some unit-level trainer programs. Therefore, our goal for cockpit geometry was that the location, appearance, and feel of all cockpit controls and displays would be the same as in the airplane.

Trainer availability was also a design objective. High fidelity is of no value if the trainer fails frequently or requires a long time to repair. The training system user availability requirements have been increasing in recent years and meeting them has become a real challenge. We established our availability goal at 98%, using the definition that the trainer would be available for training 98% of the scheduled training days each year.

Minimize Cost

We knew the cost had to be low; but just how low? We focused on keeping recurring cost in the $500K-$800K range as our target.

IDENTIFYING COST AND FIDELITY DRIVERS

Training Task Analysis

Perhaps the single most important activity in cost-effective trainer design is the front end training task analysis. This activity is used to conduct the fidelity/cost trade-off that will ultimately end up driving the design. A thorough training task analysis should be applied to all subsystems of the trainer that are used to present cues to the pilot. One of the most significant and costly subsystems in a flight simulation trainer is the visual system. This paragraph looks at the task analysis and how it was applied to the design of our trainer's visual system.

Process - The purpose of our task analysis as applied to the visual system was to determine the visual system field-of-regard (FOR) and scene content necessary to train specific F-16 pilot tasks. From a listing of all F-16 pilot tasks associated with the F-16 and its mission, 128 different tasks were identified as potentially requiring a visual presentation during flight simulation training. A sample of experienced F-16 pilots was used to plot the desired and minimum visual FOR required to train each task at a 90%, 95%, and 100% level. Each task was also rated with an importance code. The tasks were then divided into task groups (e.g., normal procedures, emergency procedures, air-to-air weapons employment) and consensus plots were then determined for each task group. The pilots were also asked to determine visual scene content requirements for each training task group by rating a list of capabilities as critical, desirable, or not needed.

Results - Figure 1 is an Aitoff plot showing the outcome of the field-of-regard analysis. The dotted line represents the minimum FOR for all training tasks trained at the 90% level (280°H x 100°V). This FOR would require approximately ten channels of video supported by a full dome. Cost: approximately $2.5M. The dashed line represents a compromise by eliminating formation and electronic combat tasks (correlating RWR signals with visual sightings) requiring a wide FOR. Although this represented the optimum visual system FOR (200°H x 100°V) for the remaining tasks, it would require six channels of video supported

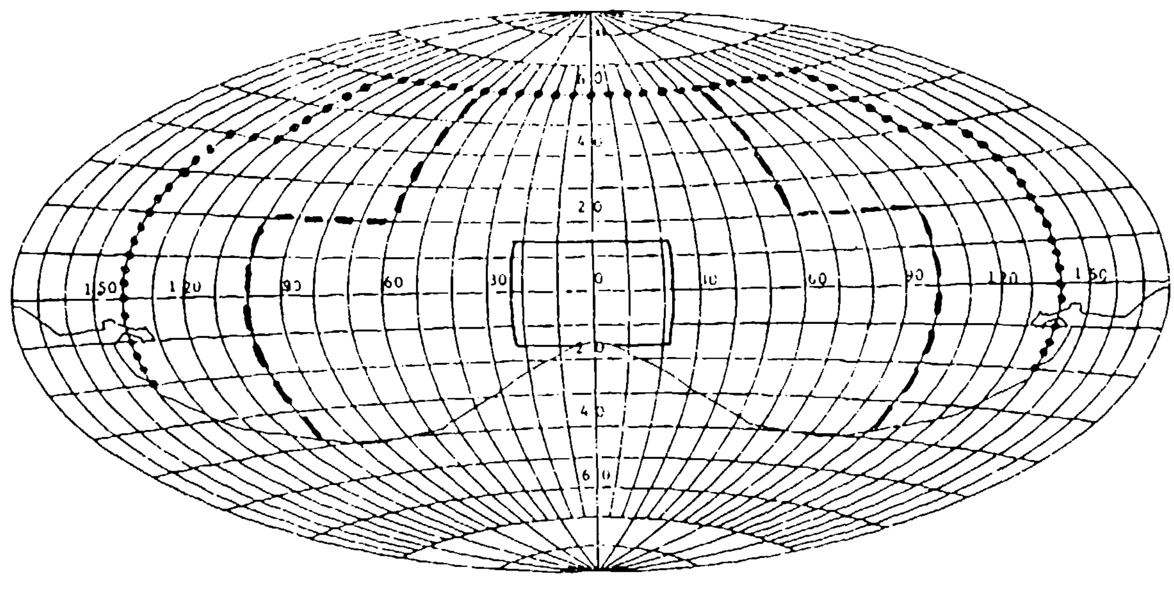


Figure 1. Aitoff Plot Showing Field of Regard Requirements

by a partial dome. Cost: approximately $1.5M. It was clear that a visual system of this size exceeded our definition of low-cost. Our final compromise is shown by the solid line representing a single channel, 45°H x 32°V system. This FOR will support the critical training tasks required of a low cost unit level trainer and represents the lower forward channel of the six channel system. Cost: approximately $250K. A key factor in our selection of this system was that any low-cost, narrow FOR solution must be able to expand gracefully to eventually accommodate the optimum FOR of 200°H x 100°V.

The outcome of the scene content capabilities assessment was somewhat surprising in that it tended to favor capabilities found in lower cost image generators as critical or desirable. Our pilot sample was of the opinion that the content and capability found only in higher end image generators were generally not needed to train the tasks in our task listing. Following is a partial summary of the scene content results:

1. Critical requirements: Horizon, Airfield, Wingman, Air Threats

2. Desired requirements: Haze, Rain, Clouds, Generic Terrain, Vegetation, Air Threat Missiles, Smoke Trails, Tracers

3. Not needed: Fog, Dust, Smoke, Sun Angle Effects, Geo-Specific Terrain, Bodies of Water, Roads, Buildings, Tanks, Trucks.

Cost Analysis

The training task analysis helps to identify the requirements which must be supported by each subsystem of an aircrew trainer. (This was illustrated in our previous example for the visual subsystem.) It addresses such issues as required aural, visual, and tactile cues and component level of fidelity. To accommodate the requirements dictated by the training task analysis in the most cost effective manner possible requires a detailed cost analysis. The cost analysis must determine the significant cost drivers within a given subsystem; identify relative costs (including life cycle costs) for alternate designs or the application of new technology, and addresses potential compromises in the defined requirements and level of fidelity identified for the subsystem.

The most challenging subsystem to effect significant cost reductions in our design studies was the trainer cockpit. Although the cockpit may not be the most costly subsystem of a typical low-cost trainer design, it is the core element which is required regardless of training application or options selected. The cockpit must faithfully represent the actual aircraft cockpit in form, fit, and function as will be shown

later. The smallest deviation (though possibly not detrimental to training) is quickly detected by the most casual of users. Therefore, any potential design concept which could help minimize costs must be weighed against its impact on cockpit fidelity and ultimately user acceptance.

Process - In our cost analysis for the cockpit subsystem, we identified four significant cost drivers listed in order of magnitude:

1) Electro-Mechanical instruments - either actual aircraft or simulated;

2) Displays - Multifunction Displays (MFDs), Radar Warning Receiver (RWR) display, Data Entry Display (DED), and pilot fault list display (PFLD);

3) Controls or control assemblies - Pilot control stick and transducer, Throttle assembly, and rudder pedal assembly; and

4) Harnesses/Cables - especially those that support the first three cost drivers.

The following list presents examples of initial procurement costs for some of the components considered in the identified cost drivers:

Electro-Mechanical Instruments

Horizontal Situation Indicator (HSI)	$25,000
Attitude Director Indicator (ADI)	$10,500
Altimeter	$ 7,000
Mach Airspeed Indicator	$ 5,200
Vertical Velocity Indicator	$ 4,200
Angle of Attack	$ 3,000
Back-up ADI	$ 3,000
	$62,100 Per Cockpit

Displays

Multifunction Displays (MFDs)	$10,000
Radar Warning Receiver (RWR)	$ 3,000
Data Entry Display (DED)	$15,000
Pilot Fault List Display (PFL)	$15,000
	$43,000 Per Cockpit

To these costs were added initial spares cost and a yearly replenishment spares cost to estimate life cycle cost for subsequent comparisons with alternate designs.

Results - In our analysis of alternative designs and low cost technologies, we determined that the application of new "glass display" technologies would have the largest single impact on the defined cost drivers. A broad range of display types was considered as potential candidates for replacing the expensive electro-mechanical instruments and displays. The question was, "Can we make application of one or more of these display types to help reduce cost and still retain the high-fidelity requirements of the cockpit?" The answer is "Yes!"

PROBLEMS AND SOLUTIONS IN BLENDING LOW-COST WITH HIGH FIDELITY

Seeking an Alternative to Expensive Mechanical Instruments and Displays

Faced with the high initial and life cycle costs associated with mechanical instruments and actual aircraft display hardware (such as multi-function displays and data entry displays), we proceeded to explore alternatives.

Process - The purpose of this phase of our design was to determine the technical feasibility of designing an F-16 cockpit trainer using "glass" components. The following current off-the-shelf glass display devices were evaluated:

- Thin-Film-Transistor (TFT) Displays
- Plasma Displays
- Light Emitting Diode (LED) Displays
- Electroluminescent (EL) Displays
- Cathode Ray Tubes (CRT)

We supported this analysis of alternative end display technologies by using existing laboratory instrument and indicator software and an existing F-16 part task trainer design as baselines. The hardware and software baselines were then modified to support the various alternative components.

Results - Results of our analysis led to the conclusion that color Cathode Ray Tubes (CRTs) offered the optimum solution for implementation of a virtual instrument display for the F-16 cockpit configuration. We found that CRT's were available off-the-shelf in a wide variety of sizes making them easily adaptable to the instrument panel geometry. Color CRT's were readily available from a wide variety of

domestic vendors reducing response time. CRT's did not have the narrow viewing angle restriction as did most other devices evaluated. This was an advantage in that the displays can be viewed by an instructor positioned at the side of the cockpit as well as the pilot in the cockpit. Finally, CRT's were the least expensive of all other display devices. Other available display components or technologies had significant shortcomings:

1. TFT displays had a poor viewing angle in both horizontal and vertical axes. Their shallow depth required the electronics to be packed into a wide frame surrounding the display glass. This compromised edge matching with contiguous displays.

2. Plasma displays represented an emerging technology. However, commercial color plasma displays were not available at the time of our study.

3 LED displays offered inadequate resolution to portray moving instruments. Only monochromatic LED displays were available.

4. EL displays were found to be monochromatic only, with wide frames. However, they were available in a wide variety of sizes.

Adapting a Virtual Instrument Display to Cockpit Geometry

Adapting a virtual (or "glass") instrument display to the F-16 cockpit proved to be a real challenge. Unlike most aircraft with rectangular instrument panels in one plane, the F-16 has a T-shaped instrument panel in multiple planes. A rather simple solution using a single, large monitor to display the instruments could be used, but we concluded the compromise to total cockpit fidelity would ultimately be unacceptable to the user. We proceeded with a design using smaller, multiple CRT's. To keep cost low, the monitors had to be commercial, off-the-shelf items.

Mockup - A proof-of-concept plywood mockup was used to evaluate various CRT types and configurations. A configuration using four small color CRT displays proved to be most compatible with the F-16 cockpit. Our mockup showed this concept would result in a trainer design with virtually no deviation from the aircraft cockpit geometry. Figure 2 shows the compatibility of the four CRT's with the F-16 instrument panel. Figure 3 shows the compatibility of the four CRT design concept with the pilot instrument line-of-sight depression angles.

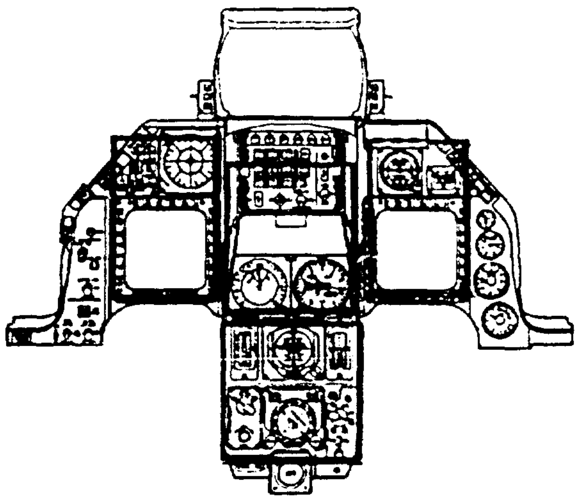


FIGURE 2
Monitor/Instrument Panel Layout

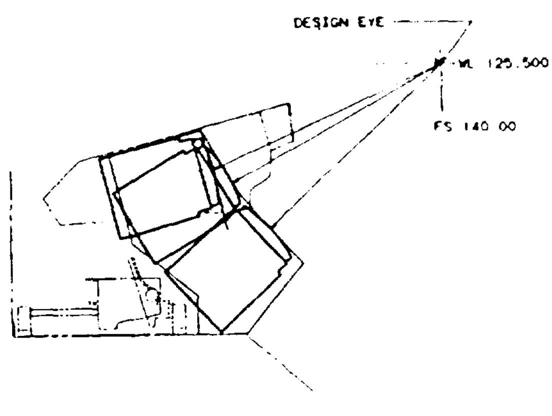


FIGURE 3
Line-of-Sight Geometry

Reducing Instrument Swimming Effect

A major criticism of glass display technology has been in the ability to adequately reconstruct the fine detail in dynamic instruments such as

the Attitude Director Indicator (ADI) and Horizontal Situation Indicator (HSI). Problems that historically arise in this application are due to achievable pixel resolution. Limitations exist either in the selected display generator and the size of the display buffer or in the resolution achievable with the selected display device. Improving achievable pixel resolution relates directly to increased cost. The problems are manifested to the observer as an inability to resolve the fine detail (alphanumerics, vectors, etc.) and/or an apparent "swimming" (aliasing) effect. This latter problem is due to pixel quantization which produces the familiar stair-step effect in diagonal lines drawn with a raster-type display generator. The search for a solution to these problems was greatly aided by the choice of display device from our analysis of alternative "glass display" technologies. The selected color CRT display is a non-interlaced VGA design (640 pixels by 480 lines) in a 4:3 aspect ratio providing approximately 90 pixels per inch resolution. This electronic resolution is supported by a display dot pitch of .01 inches and a display generator operating at VGA resolution. With this CRT display placed at the proper distance from the pilot design eye point the resolution achieved was more than adequate to resolve the detail in the instruments evaluated. The "swimming" effect was significantly minimized by this combination of pixel resolution and relative geometry. Further reduction was achieved by adjusting the brightness level of the background fill (normally black) to a barely perceptible shade of grey, and increasing the update rate of the instrument display software.

The background fill value helped soften the edges of displayed detail by effectively reducing the contrast. The increased update rate minimized the discrete steps between each rendering of the instrument face producing smoother motion. Although low cost anti-aliasing techniques involving pixel replication or bi-linear interpolation did slightly improve the swimming (aliasing) effect, the resultant loss of resolution was unacceptable. Better anti-aliasing techniques were judged to be cost and/or performance prohibitive. In summary, the proposed solution to the simulation of trainer mechanical instruments and displays has been extensively evaluated by active aircrew members and judged to be very acceptable for training including precision instrument flight tasks.

THE IMPORTANCE OF TRAINER COCKPIT FIDELITY

The challenges and solutions addressed in this paper focus primarily on reducing cost without compromising cockpit fidelity. More specifically, they focus on the accurate replication of the trainer cockpit geometry. The practical benefit of achieving accurate trainer cockpit geometry is to gain long term user acceptance of the device. Military aircrews seem to accept avionics familiarization trainers and procedural trainers that violate cockpit geometry. As soon as the familiarization training is over and the procedural tasks learned, these lesser fidelity trainers are relegated to the squadron storage room. But when the training device is designed to simulate flight, that's when the typical aircrew will demand high fidelity cockpit geometry.

Design Eyepoint

Airplane cockpits are designed based on a design eyepoint - the eye location of the mythical "90% man" when sitting in the cockpit. Human factors engineers make careers out of designing ergonomically efficient cockpits. Every display and control has its position based on the airplane design eyepoint. For example, in fighter aircraft, the head-up display (HUD) plays a critical role in weapons delivery. Indeed, some pilots consider the HUD to be the primary instrument in the F-16. The HUD's display and functionality are based on the design eyepoint. The airplane boresight, traditionally used as a backup weapons delivery mode, is based on viewing from the design eyepoint through the HUD. The flight path marker, a HUD symbol displaying the airplane's path through the air, must be viewed from the design eyepoint.

The flight instruments are positioned and organized from the design eyepoint in a way that supports efficient viewing by the pilot. The pilot's "instrument scan pattern" is a behavior he develops over many hours of flying a particular airplane. This behavior becomes second nature to the point that he does it without thinking. A cockpit trainer which simulates flight must support this critical learned behavior pattern. To do so, the trainer's cockpit geometry must be based on the airplane's design eyepoint. Figure 3 shows the F-16 instrument panel line-of-sight depression from the design eyepoint. The pilot's

outside visual FOR is based on the airplane's design eyepoint. A cockpit trainer's visual simulation FOR must also be based from the design eyepoint. For example, the pilot's line-of-sight (LOS) over the nose and canopy rail is critical for weapons delivery and landing training. If a trainer does not accurately replicate LOS geometry, it will violate previously learned behavior patterns. History shows that any cockpit trainer which is designed to simulate flight, but which violates the concept of design eye geometry, is doomed to controversy and, ultimately, rejection by the user.

Pilot's Seat

The pilot's seat in the cockpit trainer is inextricably linked to the design eyepoint, and is, therefore, just as critical in trainer design. Our experience indicates that a pilot's first act on sitting in a cockpit trainer is to adjust the seat. What he is subconsciously doing is positioning his eyes at the aircraft design eyepoint (or his personally established deviation relative to the design eyepoint). We concluded that an accurate replication of the aircraft seat, position, inclination, and adjustment envelope was just as critical as design eye geometry in a low-cost trainer. Why design the trainer cockpit based on design eye geometry if the pilot can't position his eyes to his customary location relative to the design eye? Accurate geometric replication of the seat functional controls was necessary to support emergency procedures. Accurate location and feel of the ejection handle and inertial reel locking lever were necessary to support virtually unconscious behaviors in emergency conditions.

Stick, Throttle, and Rudder Controls

The stick and throttle in today's high performance fighter aircraft do much more than move the flight controls and change engine thrust. They also are used to control weapons employment, avionics function, and communications. The days of a simple pickle button on the stick and radio mike button on the throttle are long gone. For example, the Block 50 F-16 stick and throttle grips have a total of 16 multiple position switches. These switches are designed to be selected by tactile identification without distracting from the pilot's visual tasks. Accurate tactile fidelity and geometric replication of the stick and throttle were critical design criteria for our trainer.

Our design analysis showed that rudder pedal form and function were also critical to our design. While the rudder is seldom used in routine F-16 flight, its use is required for takeoff, landing, and some emergency procedures training. Pilot input to the design required that we provide a rudder pedal adjust mechanism to permit full rudder pedal movement and authority.

Instrument-Mounted Switches and Controls

As our glass virtual instrument display design matured, an associated problem evolved which directly impacted cockpit fidelity and functionality. The problem was how to accommodate instrument-mounted switches and controls on a glass-faced CRT! Our research into user requirements found that it would be unacceptable for the pilot to reach outside the trainer cockpit to make control inputs that would be made on the instrument panel in the airplane. For example, a rotary control device on the face of the HSI is used to set the desired course on the instrument. Feedback from users indicated it would not be acceptable for the pilot to make the HSI course adjustment by making an input through a trainer control panel outside the cockpit.

A solution to this problem was found by designing thin form-fitting aluminum bezels that overlaid the CRTs. The bezels accurately depict and simulate the forward instrument console. The bezels incorporate very low profile controls to provide normal instrument control functions. The bezel overlays swing away from the CRTs to facilitate maintenance. Figure 4 shows two of the bezels swung down to reveal the CRTs.

PRACTICAL BENEFITS OF THESE SOLUTIONS

The use of the alternative virtual instrument display system to replace electro-mechanical instruments and cockpit display hardware is a good example application of a low-cost technology. But what are the real benefits of this exercise? Does it just produce another novel cockpit trainer design? We identified three categories that would help establish comparisons between the new cockpit design approach and a classical approach. These categories are cost (initial and life cycle), reliability, and configurability.

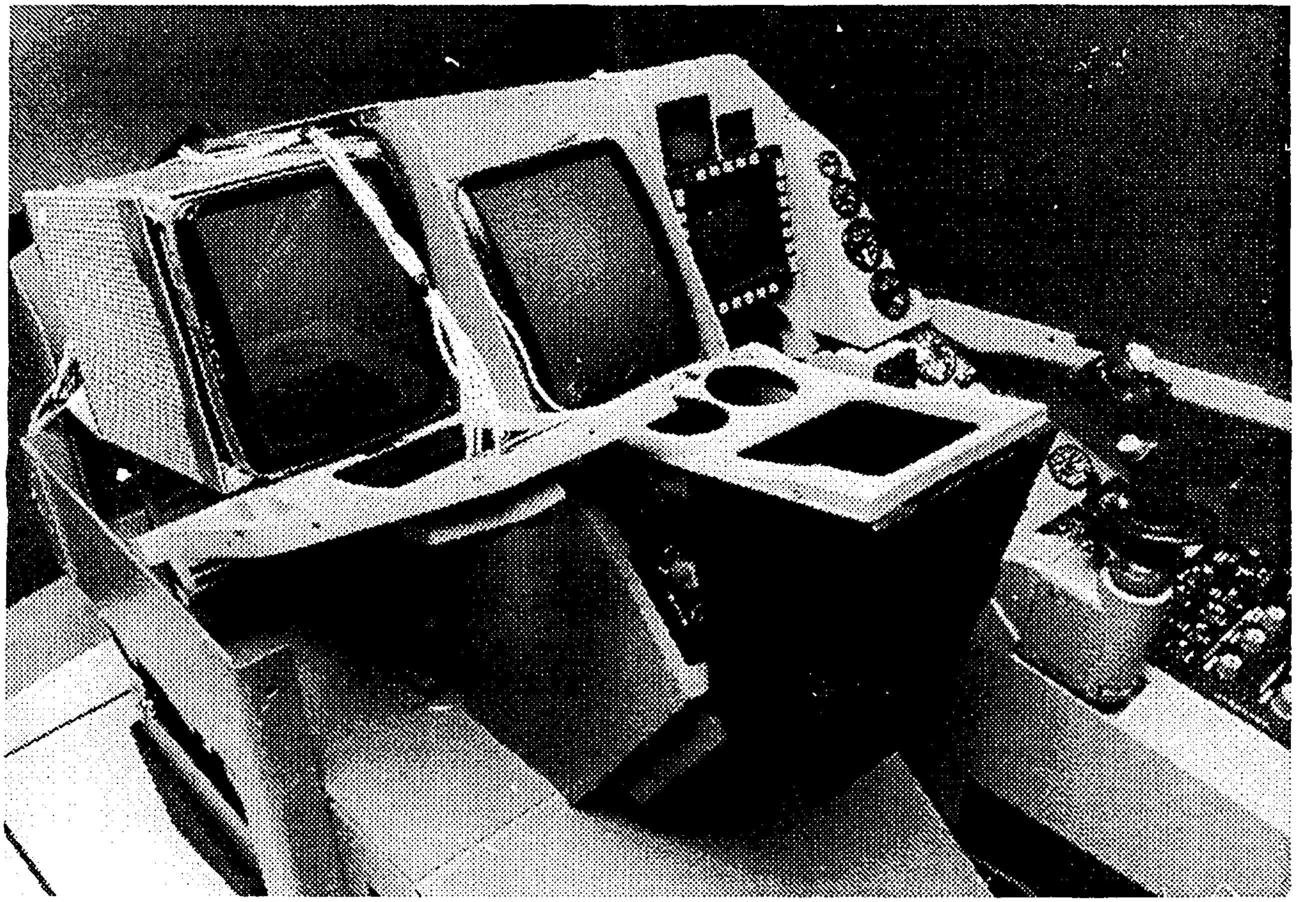


FIGURE 4 - Instrument Panel Bezels Swung Down

Cost Comparison

A cost comparison based on 10 cockpits over a 10-year life cycle was conducted between the two design approaches. All component and fabrication costs for an initial buy were well established using actual published off-the-shelf prices for procured components and actual fabrication costs for accommodating the new virtual display hardware. Costs for initial spares and replenishment spares were estimated based on historical data. The result of the cost comparison is summarized in Figure 5.

Reliability Comparison

The reliability comparison is based on published mean time between failure (MTBF) data for the major components and a measure of the relative complexity between the two designs. The virtual display design eliminates over 40% of the electrical and mechanical components and cables associated with a typical trainer cockpit. Both measures indicate a higher reliability for the virtual instrument display design over the classical design. The results of these comparisons are summarized on Figure 6.

Cost Comparison (10 cockpits over 10-year period)

	Virtual Display	Classical	Delta
Initial Cost (per cockpit)	$34,600	$71,407	$36,870
Initial Spares (per cockpit)	$32,500	$61,617	$29,117
Replenishment Spares/Yr (based on 15% factor)	$4,875	$9,243	$4,368
10 Cockpits over 10-years	$1,158,500	$2,254,540	$1,096,040

Net Savings:

Initial Buy - $660K
Out-Year Spares - $436K
Total Life Cycle Savings - $1.1M

FIGURE 5 - The Virtual Instrument Display System Shows a Cost Savings Benefit

	Virtual Display		Classical
Relative Complexity	4 Color CRTs 4 Video Cables	replaces	14 Instruments-Displays
	12 Wires	replaces	9 Harnesses (200 wires)
Major Component MTBF	- Color CRT (100,000 Hrs.)		- Simulated HSI (8,000 Hrs.) - Simulated ADI (10,000 Hrs.) - Simulated MFD (23,000 Hrs. estimated) - Simulated Altimeter (20,000 Hrs.) - Simulated Mach Airspeed (25,000 Hrs.)

FIGURE 6 - The Virtual Instrument Display System Indicates a Reliability Benefit

Configurability Comparison

The virtual instrument display system provided the additional benefit of configurability due to the flexibility in the multi-CRT design and bezel concept and the software controllability of the displays. The differences in instrument console geometry, such as exists between the F-16A model and F-16C model cockpits can be accommodated by proper placement of the four-CRT array and appropriate bezel design. Different instrumentation in the cockpit, such as a needle and dial Vertical Velocity Indicator (VVI) versus a tape VVI can be supported through a simple software change. Similar modifications to a classical cockpit would require not only extensive hardware design modification, but numerous changes to the instrument suite, display suite and computer/cockpit interface.

CONCLUSION

This paper summarizes three years of data acquisition, analysis, and design. Our overall objective of this activity was to determine the feasibility of providing the military customer low-cost, high-fidelity flight simulation training in the near future. We have successfully demonstrated our specific project objectives of maximizing fidelity and availability, and minimizing cost in a unit-level trainer. In addition, we realized another benefit of achieving system flexibility through reconfigurability. We have concluded that the future of low-cost, edge-of-the-envelope, flight simulation training for the military aircrew is available by way of this described approach.

DARTS:
A DOMAIN ARCHITECTURE FOR REUSE IN TRAINING SYSTEMS

Robert G. Crispen, Brett W. Freemon, K. C. King, and William V. Tucker
Boeing Defense & Space Group
Huntsville, Alabama

ABSTRACT

The dynamics involved in the training system marketplace of today are dictating the need for major changes in the way organizations specify, develop, and maintain training systems. One of the key areas affected by these changes is the system and software architecture of training systems. This is evidenced by the increased attention that has been placed on architectures by recent initiatives (e.g. Structural Model, Mod Sim, STARS, DIS, ARPA DSSA, etc.). There are many reasons for this emphasis, not the least of which is a desire to produce training systems at the least possible cost while providing faster time to market and higher quality. An architecture for training systems can be a framework to enable cost reduction, reusability, and standardization.

We derive a set of attributes which we believe characterize a "good" software architecture. We discuss an architecture developed by Boeing Defense & Space Group, the Domain Architecture for Reuse in Training Systems (DARTS) and evaluate DARTS against these criteria. We also discuss the role of DARTS in megaprogramming, part of the ARPA STARS initiative, and suggest that DARTS is a suitable architecture for achieving the STARS vision of process-driven reuse.

ABOUT THE AUTHORS

Robert G. Crispen is a Systems Analyst with the Missiles & Space Division of the Boeing Defense & Space Group. He worked on the Modular Simulator Program, the Ada Simulation Validation Program, and is currently a researcher in an R&D program at Boeing. Before joining Boeing, he was a Senior Systems Design Engineer on commercial flight simulators at GMI in Tulsa, OK. He holds a Bachelor of Arts degree from the Johns Hopkins University in Liberal Arts/Psychology.

Brett W. Freemon is a Senior Software Engineer with the Missiles & Space Division of the Boeing Defense & Space Group. He has worked on all areas of the life cycle of simulation and training systems from proposal through delivery and installation. He is currently working on a research and development program centering on software reuse technology. He holds a Bachelor of Science degree from the Georgia Institute of Technology in Applied Physics.

K. C. King is manager of the ARPA STARS demonstration with the Boeing Defense & Space Group. This demonstration entails using STARS and megaprogramming technologies to develop an operational flight instrument trainer for the Navy T-34C aircraft. Prior to joining the STARS program, he developed architectures for a number of large-scale DoD information systems. He holds a Bachelor of Arts degree from the University of Michigan in Political Science.

William V. Tucker is the R&D Manager for Boeing Defense & Space Group, Simulation and Training Systems organization. He has managed various trainer programs, including US, UK, and RSAF E-3, KC-135 and the Modular Simulator Design Program. He is a member of a functional oversight committee for the ARPA/Navy T-34 FIT STARS demonstration project. He holds a Bachelor of Science degree from Wichita State University in Electrical Engineering.

DARTS: A DOMAIN ARCHITECTURE FOR REUSE IN TRAINING SYSTEMS

Robert G. Crispen, Brett W. Freemon, K. C. King, and William V. Tucker
Boeing Defense & Space Group
Huntsville, Alabama

WHAT IS AN ARCHITECTURE?

An architecture, as we intend to use the term, consists of (a) a partitioning strategy and (b) a coordination strategy. The partitioning strategy leads to dividing the entire system into discrete, non-overlapping parts or components. The coordination strategy leads to explicitly defined interfaces between those parts.

These two strategies provide an engineering approach to bridging the gap between the system as a whole (as represented by its specification) and the design (the plan to build the product from primitive parts, such as computer instructions, metal struts, and switches).

The reach of an architecture can extend from a single system (an architecture that solves a unique product problem) to an entire family or product line of systems. In the latter case, once the partitioning methods and coordination rules are determined, multiple products can be generated using the same methods and rules.

By the definition we are offering, the Software Engineering Institute's (SEI's) Air Vehicle Structural Model (AVSM)[1,2] and the HAVE MODULE Modular Simulator (Mod Sim)[3] fit into our discussion of architectures.

Because the architecture will determine both the list of parts for a particular product and the coordination between those parts (their size and shape, among other things) the architecture chosen for a particular training system has a decisive impact on reuse. We have observed[4] that software parts which were developed under one architecture were adapted only with great difficulty to serve in another architecture. When the architectures are significantly different, we have concluded that redevelopment is more cost-effective than re-coding.[5]

WHAT IS A "GOOD" ARCHITECTURE

If architectures are different from one another, it ought to be possible to say that one architecture is better or worse than another in some meaningful sense. Nevertheless, we have seen very little in either the training systems literature or the software engineering literature on what makes one architecture better than another. There are assertions that one architecture or another is a "good thing," but there is little public scrutiny of criteria.

We will offer the following characteristics against which software architectures can be measured. Since we are about to describe a specific software architecture, the Domain Architecture for Reuse in Training Systems (DARTS), these criteria may be unconsciously biased toward DARTS. However, we have attempted to establish a widely acceptable set of criteria.

A good architecture can be leveraged. It must show promise of lasting beyond present programs, rather than being a quick fix of specific current problems. It must be adaptable to easily fit many development methods. It must promote the highest levels of reuse maturity. It must hold up to changing requirements. And it must be scalable across a significant portion of the training systems domain.

A good architecture promotes system understanding. It must "look like" the problem space in some significant sense. It must be clear, and it must clearly meet both user and end customer requirements. Its quality and style should match what are considered sound systems and software engineering principles.

A good architecture is rational. It should promote and support a repeatable and improvable process for building out a specific member of the family.

A good architecture is affordable. It must be

"efficient enough" in both time and memory. It must support large-scale cost and schedule improvements in both the short term and the long term. And it must have been defined, published, and demonstrated to work in order to reduce risk.

A good architecture is a good citizen. It should not violate company or customer standards. It should be broadly accepted or acceptable in the training systems and customer community. It should be available in the public domain rather than being bound to a proprietary hardware or software system. It should meet the emerging framework criteria articulated by the ARPA DSSA project. And it should take advantage of military and international standards like the Ada programming language and ISO communications protocols.

HISTORY OF DARTS

Boeing Defense & Space Group had participated in the Ada Simulation Validation Program (ASVP) where the term "structural model" was first introduced to industry. We had also participated in Mod Sim[3] from the beginning of the program through our role as the prime contractor for the demonstration/validation phase. When we had a preview of structural modeling at the SEI, we were anxious to reconcile the two, if that were possible.

Fortunately, ASVP provided a common starting point.

With help from the SEI, we were able to realize what it was that we needed to develop: an architecture which captured both the reusable form of a structural model and the reusable content of Mod Sim.

The resulting software architecture, DARTS, was developed as the domain-specific software architecture (DSSA) for the Air Vehicle Training System (AVTS) used by the Navy/STARS 1993 demonstration of the benefits of mega-programming.

Characteristics Adapted From Mod Sim:

Several features of the Mod Sim architecture are incorporated in DARTS:

- DARTS is based on the notion of a generic flight simulator that is capable of being adapted into any present or foreseeable training simulator. The generic simulator is partitioned into approximately 125 air vehicle Functions or areas of capability.

- Each of these Functions is assigned to one of twelve Segments (see Figure 1).
- A Segment is characterized as being coherent internally and loosely coupled externally. That is, the Functions assigned to a Segment "go together" in the sense that there are data flow, execution order, or other dependencies between them. Functions assigned to different Segments, on the

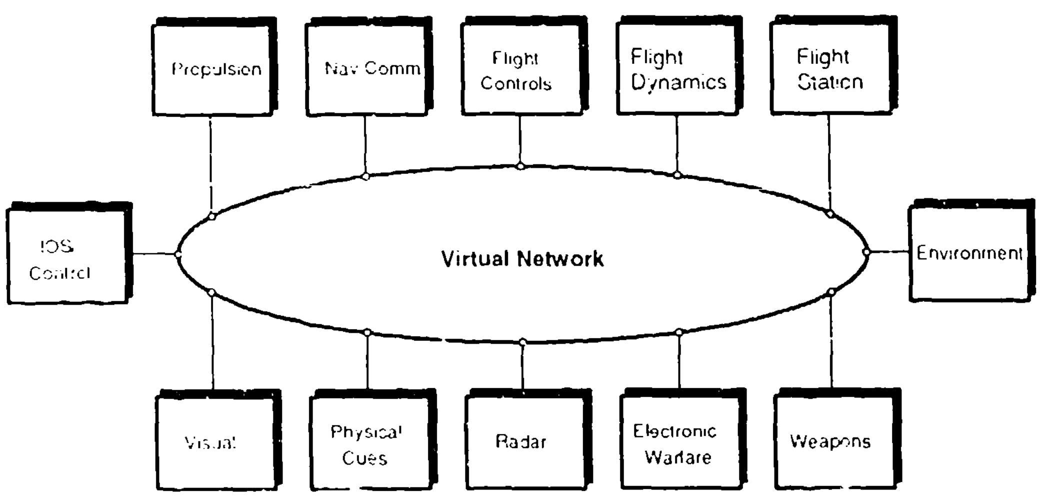


Figure 1 DARTS Segments

other hand, do not "go together" in this sense and may execute independently of Functions in other Segments, except that they may produce data for or consume data from those Functions

● Segments have a clearly defined set of interfaces with one another. The generic interface definitions, which are maintained as compilable Ada code and which are adaptable to the requirements of a specific training system, define the only means by which Segments may communicate with one another.

To summarize the first four points, a sizable block of systems engineering work was done on Mod Sim and its follow-ons. This work was subjected to an industry-wide review process and evaluated via a demonstration project. There is a defined process for adapting this work to virtually all kinds of simulators

In our conversations with the Software Productivity Consortium (SPC) and STARS, it became clear that this reusable systems engineering and its work products are very similar to the processes we now call Domain Engineering[6].

● DARTS retains the message-based communication method between Segments pioneered on Mod Sim. Message-based communications have a certain safety factor which is absent in shared-memory architectures. Imagine the following scenario: variable **X** is computed by a Segment running in another CPU. Your Segment executes the following code.

```
Y := Shared_Memory.X;
Do_Something_Else;
Z := Shared_Memory.X;
if (Y /= Z) then
  Strike_Pilot_Repeatedly_
  With_Control_Column;
end if;
```

In a shared-memory architecture, if the other Segment changes variable **X** while this Segment is `Do[ing]_Something_Else`, the pilot might become unhappy! In a message-based architecture like DARTS, on the other hand, variables are only updated when the application program requests that they be updated.

Message-based communication is not the only safe mechanism for avoiding the situation above. The SEI's AVSM, for example, has a data synchronization mechanism at the start of each frame which accomplishes the same thing. Nevertheless, shared memory often ties the builder of a training system to one or a handful of vendors of shared memory hardware. We believe message-based communication is a more general solution. We also understand that achieving universal agreement on this point is unlikely.

Mod Sim Characteristics Discarded in DARTS

A few features of the original Mod Sim architecture imposed unnecessary restrictions on implementors, and were replaced in DARTS:

● Basic Mod Sim divided a trainer into twelve separate boxes, called Modules. Because DARTS ought to work on *any* computer system or combination of computer systems, we discarded the notion of one Segment per box (or Module). Instead, we distinguished between Segments (closely coupled software systems) and Modules (computational systems) so that any number of Segments could reside in a Module. The current version of the Mod Sim specifications have adopted this convention as well.

● Mod Sim Segments communicated with one another over a fiber-optic network. This capability is still available for communication between Segments which reside in different Modules, but for Segments in the same Module, it makes sense to communicate through shared memory.

The wrong way to do this is to require individual Segments to know where they are and where other Segments are, so that they can use shared memory for some communications and fiber optics for others. The right way, in our opinion, is to create the concept of a Virtual Network (VNET). A Segment simply calls "Put" or "Get" indicating that it wants to give data to other Segments or get data from other Segments. It is up to the VNET software to determine how to transfer the data.

Characteristics Adapted From Structural Model

The internal workings of a given Segment are irrelevant to the operation of a "classic" Mod Sim. So long as the Segments meet their interface requirements, any internal architecture can be used.

Note that requiring standard interfaces is not the issue. Having standard interfaces makes it quite simple, for example, to subcontract and subsequently accept one or more Segments on the simulator. In the Mod Sim demonstration, we produced only two of the eleven Segments, subcontracting the other nine. This capability was retained in DARTS.

But to assert that the internal workings of a Segment don't matter is to assert that any architecture is as good as any other, which is contrary to our thesis. Accordingly, with only a few modifications that we describe below, we incorporated the SEI's AVSM into our Modules and Segments.

THE DARTS ARCHITECTURE

An overview of DARTS is shown in Figure 2. A training system is divided into Segments; Segments are divided into Subsystems; and Subsystems are divided into Components. Segments are grouped together into Modules. Note that the analysis that produces the final architecture begins with functional decomposition and ends with what can sensibly be described as objects.

Is the DARTS analysis methodology "real" Object Oriented Design (OOD)? In one sense it certainly is, since the leaf nodes are what anyone would describe as "objects". On the other hand, since DARTS begins with a functional decomposition, it is occasionally necessary to divide the function of a Component into several Segments. For example, a hydraulic pump Component may exist in the Flight Station Segment to produce the simulation of hydraulic fluid flows, while a hydraulic pump Component may also be required in the Physical Cues Segment whose only function is to simulate the sound of the pump's operation.

We have preferred to use the term "Object Abstracted Design," and we follow those who view the applicability of pure OOD to training systems design with some skepticism[7]. Rich McCabe of the Software Productivity Consortium gave an insight which may temper some of the passions aroused by this issue: as a rule, functions in the real world are accomplished, not by spirits or demons, but by objects.[8] To found a systems engineering practice on this commonsense notion seems at least defensible.

Module Executive

There is one Module Executive for every Module (computational system). All operating system and hardware dependent functions such as interrupt, task suspend and resume, and so on, are located here. The Module Executive "causes" the Segment Executives to execute. This is kept deliberately ambiguous, since the right way of doing this on a given program may be to call the Segment Executives as subprograms, or it may be to schedule their execution as independent tasks. Because data flow between the Module Executive and Segment Executives is one-way and small (the clock tick message passes from the Module Executive to its Segment Executives), it does not stand in the way of implementing the right choice for a program.

Segment Executive

The Segment Executives are responsible for all communications over the VNET apart from the clock tick message. By isolating the VNET communications functions in the Segment Executives, the lower-level elements (Subsystem Controller and Component) may be reused from similar software for other architectures. All data contained in messages (that is, all data defined in the adapted DARTS Interface Specifications) flows through the Segment Executives between the VNET and lower level elements.

The Segment Executives are also responsible for mode and state control logic (total freeze, reposition, run mode, and so on).

The Segment Executives schedule the execution of their Subsystem Controllers by using a scheduling table mechanism similar to that used in the AVSM. A difference between DARTS and the

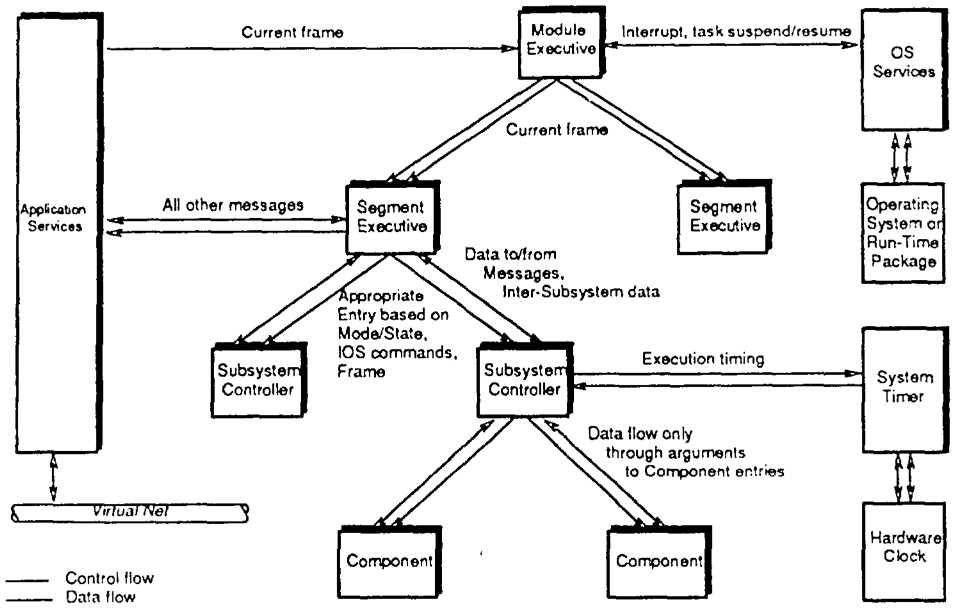


Figure 2. Domain Architecture for Reuse in Training Systems Overview

AVSM is that functions such as malfunction insertion and mode/state change which the AVSM handles through a separate aperiodic scheduling thread are handled in the main execution thread in DARTS: a mode or state change message in DARTS is a message like any other, though it is processed by the Segment Executives. Because of the way the AVSM does aperiodic execution, this is not a large or significant change.

The Segment Executives in DARTS call upon the appropriate aperiodic entries in each of the Subsystem Controllers, based on the receipt of the appropriate control messages through the VNET.

Subsystems and Subsystem Controllers

Subsystems correspond to the Functions allocated to Segments in the Mod Sim architecture. Though the analysis has been completed on only half the Segments so far, it appears that there will be little difficulty in accomplishing this for other Segments. Nevertheless, the possibility must be raised that more than one Subsystem will be required to implement a given Function. There is no structural impediment to doing this in DARTS.

Subsystem Controllers are implemented as in the AVSM. In the AVSM data flows out of Subsystems through a shared memory based Export Area, while in DARTS (largely because of the correspondence between Subsystems and interface messages) the Segment Executive provides the Subsystem Controller with data from messages and builds messages to send to the VNET.

All data flow between Subsystems takes place through buffers maintained in the Segment Executives.

We believe that it is the responsibility of the individual Components to provide "safe" input values for themselves. Thus, the Initialize entry for each Component provides both input and output data. Once the Component has executed its Initialize function, it may execute without error even though no input data has yet arrived over the VNET. This eliminates all worries about which Components or Subsystems need to execute before others in order to avoid erroneous data being processed.

The VNET provides information to the Import function as to whether or not new data has been received since the last iteration. DARTS takes advantage of this by only copying data from message buffers when new data has been received. This new-data information is not available in the AVSM.

Components

As in the AVSM, the lowest level element is called the Component. Each of the Components corresponds to an Object in the OOD sense. Thus, a Hydraulics Subsystem may consist of Pump, Valve and Reservoir Components. However, the Hydraulics Subsystem may contain Components such as Flows, Bleeds and Pressures which are less clearly objects. Nevertheless, these "function objects" are assigned to Components so that the Subsystem Controller only needs to contain "glue" logic between the Components

In DARTS, as in the AVSM, all knowledge about the operation and state of Components is contained within the Components. And no knowledge about the external environment (simulation control commands, presence or absence of other Components, computational environment) is contained within the Components. Components compute the state of the objects they simulate in a purely abstract, and therefore reusable, manner. These rules, which are among the most attractive features of the Structural Model, comprise "knowledge firewalls".
Just as in the AVSM, all data flow between Components takes place through the subprogram calls for each of the entries in the Components. As the SEI points out, this set of entries is both necessary and sufficient to permit the knowledge firewalls described above to operate.

MEGAPROGRAMMING AND DARTS

A team has been assembled consisting of ARPA, NAVAIR, NTSC, Boeing, DUAL Inc., and the SPC to demonstrate the applicability of the concept of megaprogramming to the training system domain, and, as part of the process, to evaluate DARTS as a domain architecture for achieving reuse in the context of megaprogramming.

As defined by STARS, megaprogramming is "the practice of building and evolving computer software component by component. Megaprogramming builds on the processes and technologies of software reuse, software engineering environments, software architecture engineering, and application generation in order to provide a component-oriented product line"[9].

To realize a quantum improvement in the way software-intensive systems are developed, megaprogramming envisions two distinct but cooperating lifecycles, corresponding to the family of systems (product line, domain) and to the specific system (product) respectively.

Architecture becomes a key unifying feature of the product line lifecycle, while processes for its use are the driver for the product or project lifecycle.

The processes which drive the product line lifecycle are collectively known as domain engineering and include not only the familiar notion of domain analysis, but extend to managing the product line investment, creating reusable assets (processes and components) and supporting multiple projects that use those domain assets.

Under megaprogramming, the process of building a specific system is referred to as application engineering. Achieving the quantum improvement expected by megaprogramming comes primarily through leveraging the processes, components, and technology assets developed under the domain engineering investment effort to produce individual products very uniformly, quickly, and at the lowest cost per product.

The heart of this investment in domain assets is to pre-position all of the commonality among members of the family along with processes for adding values for the defined variability among all possible members of the family. For example, all Operational Flight Trainers simulate aircraft engines, while the number of engines varies from aircraft to aircraft.

Domain engineering work products are being developed for the Air Vehicle Training System (AVTS) domain based on the DARTS architecture. The work products follow the SPC Synthesis guidelines and are derived from the DARTS architecture.

Each of the DARTS Segments has been defined as a domain and specified with: (a) a decision model for capturing the variability of the domain; (b) product requirements for representing the adaptable requirements; (c) product design for representing the tailorable design data; and (d) process specification that guides the application engineer through the instantiation of an instance of the domain.

The final step of the domain engineering process is to implement the domain (i.e., adaptable code and documents and information for their generation) so that the application engineer can generate the products for a given program.

Within the domain of each Segment, DARTS guided the domain analysis and each of the work products. Generally, Functions re-used from Mod Sim became DARTS Subsystems, and Components were derived for each of the Subsystems.

These work products are being incorporated into a Software Engineering Environment that will be used by application engineers to construct a T-34C Flight Instrument Trainer (FIT). The primary purpose of this demonstration effort is to show the benefits of megaprogramming on a real-world air vehicle training device.

DARTS support of and conformance to megaprogramming has been recognized in its adoption by the Navy/STARS demonstration project. DARTS is specific to a domain; in this case, the product line of air vehicle training systems. With sponsorship from ARPA, the engineering data foundation of DARTS is being validated by subjecting it to a formal, defined domain engineering process authored by the SPC.[6] Using the SPC's domain engineering process, DARTS is being configured to support high leverage reuse in the form of domain commonality and variability. Again, using the SPC's domain engineering methodology, DARTS is being extended to include defined processes for building out any member of the air vehicle training system product line.

ADVANTAGES OF DARTS

The performance of DARTS, as it appears to us at the present time, against the criteria we discussed at the start of this paper is summarized in Figure 3. Some advantages of DARTS which were captured from its progenitors deserve special mention.

Advantages Captured From the AVSM

The first set of advantages of the DARTS follows the advantages given for the AVSM, because DARTS incorporates such large parts of the Structural Model.

- The Subsystem Controllers and Components are based on reusable templates. Every Subsystem looks like every other Subsystem and every Component looks like every other Component, in that they have the same subprogram entries and the same package structure.

- Components are so structured as to be widely reusable. Since Components have no knowledge of their environments and little dependence on the architecture, they should be reusable in the widest possible context.

- Reuse becomes a matter of selecting and adapting from this set of identical parts, and it is entirely possible to automate this selection and adaptation based on a decision model captured in a SEE.

Leverage
- Major elements proven on multiple programs (F-16 Mod Sim, USAF Structural Model)
- Not tied to any CASE tool or computer vendor
- Subsystem specs and bodies and Component specs may be automatically generated
- Reuse of Components across multiple architectures
- Scalability by plug-replacement of Components, Subsystems, Segments
- Segments may be eliminated or combined for product-line variations
- Based on industry-wide Domain Engineering effort

Simplifies System Understanding
- Small number of well-defined elements (12 Segments, Subsystems, Components)
- Structure maps to requirements analysis (early management visibility into software)
- Software engineering principles from SEI, SPC and STARS

Rational
- Subsystems and Components are a toolset, not a straitjacket
- Results of early systems engineering activities flow into design
- Interface specifications clarify requirements, guide design

Affordable
- Much systems engineering work is already done, simply by selecting the architecture
- Parallel development, testing improve schedule performance
- Lower integration time proven in F-16 simulator program
- Architecture proven in F-16 program to be fast, cheap enough for 50 Hz WST
- Exact specification of computer power, best computer architecture for segment

Good Citizen
- Based on standards in public domain (FDDI, XTP, Ada)
- Mod Sim and Structural Model government-sponsored for simulation industry
- ISWG got wide consensus from government, simulation industry
- Contact with ARPA DSSA program through STARS

Figure 3. Summary of DARTS Performance Issues

● DARTS provides an integration harness for each of the Components and Subsystems that can permit early, structured testing. Integration of Components into working Subsystems, integration of Subsystems into working Segments, and integration of working Segments into a working training system can be done in a structured manner, and early prototyping steps of design can be done with actual components.

Advantages Captured From Mod Sim

The second set of advantages of DARTS is derived from the advantages in the Mod Sim architecture, because many of its strongest features are also incorporated in DARTS.

● Since DARTS begins with a widely accepted decomposition based on functional requirements, requirements traceability is illuminated rather than obscured by the architecture.

● The division of Functions into Segments facilitates scalability. Quite often, the functionality of entire Segments is not required for a given training system. For example, Electronic Warfare is not commonly found on transport aircraft simulators.

● Delivery is more predictable, since the components are all nameable and locatable very early in the program. Each of the Subsystems and Components can be tracked from a very early date in the program, so that reaction to delays and data voids have lower impact.

● DARTS is designed to permit Segments to be easily subcontractable. Companies with expertise in visual systems, electronic warfare, or weapons but which have little or no training system experience can compete to build the appropriate Segments. Software development and testing can take place in parallel with many workers and organizations until the very latest point in the schedule, thus greatly reducing time to delivery. Further, the ability of Segments to be tested as stand-alone components lowers both prime and subcontractor risk at acceptance.

● As requirements change, within a range of simulators, or as follow-ons require more or less CPU power, DARTS permits near-zero-effort addition or deletion of Segments and of computational power allocated to a Segment. Segments may be moved from one Module to another, again with near-zero effort. When this change is anticipated, a hardware architecture can be chosen for the affected Segments that permits the simple plug-replacement of CPUs with less powerful or more powerful CPUs.

● Interfaces between Segments are strictly specified in compilable Ada. Adaptation of these reusable interfaces to the requirements of a specific program is accomplished by the decision model for the domain, and we have demonstrated that it is easy to automate this process.

DISADVANTAGES OF DARTS

● Communications between Segments using messages and a VNET may take a larger amount of execution time than communications using shared memory, even when data synchronization (as in the AVSM) is taken into account. We believe that this price is small on today's CPUs, and will be smaller in tomorrow's CPUs.

Futher, because messages permit a wider variety of computational hardware to be used on a simulator, the dollars-and-cents cost of computers may not be very different between the two methods. Nevertheless, we see the possibility that another architecture or a modified DARTS will be more appropriate for some programs.

● DARTS, like the AVSM, absolutely requires data flow control, which takes engineering hours. Our slogan has been, "If you want to control data flow, you've got to control data flow." The generic, adatable interface specification provides a great deal of help in this control process, and utilities associated with DARTS automate much of the tedious coding work like message setup and connection.

● Organizations and companies in the simulation industry have historically seen Mod Sim as a hardware architecture, of importance only in a niche. This Mod Sim ancestry may be a political disadvantage when dealing with those organizations.

● As we indicated earlier, since DARTS begins with functional decomposition, the functionality of some components may be spread across several Segments. We have observed that cases like this are the exception, not the rule. Still, a certain

amount of object elegance is lost in DARTS, and effort is required to maintain concurrency among these objects. We have tentatively opted for giving identical names to Components whose "pure object" functionality is divided across Segments (i.e., a Component named Hydraulic_Pump in both Flight Station and Physical Cues Segments), since most configuration management systems can be made to indicate the connection.

CONCLUSION

DARTS has been the result of an evolutionary process which has incorporated the results of current research in software engineering and principles from Boeing Defense & Space Group's experience in the simulation industry.

Research at Boeing and at the SEI has proven that the two major elements of DARTS (Mod Sim and the Structural Model) are effective, low-risk architectures which can have significant impact on program cost and schedule. Research under STARS has confirmed this.

The overwhelming advantage of DARTS becomes apparent when it is used as a foundation for megaprogramming. Given the requirements for a specific program, the decision model for the domain and the adaptable, reusable components and interface specifications of DARTS, one can indeed "turn the crank" of a definable, automatable process to produce code and documents. This vision of process-driven reuse has been realized in our work for STARS, and is so much more powerful than other reuse models, that we believe it does represent a quantum improvement in the way we develop and reuse software.

REFERENCES

1. Software Engineering Institute. "An Introduction to Structural Models", 14th I/ITSEC, Nov. 1992.

2. Software Engineering Institute. Structural Modeling Guidebook (Draft), January 1993.

3. The Boeing Company. System Segment Specification for the Generic Modular Simulator System. Volumes I-XIV, S495-10400, June 1991.

4. Freemon, Brett W. and Crispen, R. G. "Testing A Technology For Reuse", 14th I/ITSEC Proceedings, Nov. 1992.

5. The Boeing Company. Ada Simulator Validation Program Final Report, D495-49506-1, Sept. 1988.

6. Software Productivity Consortium. Synthesis Guidebook Volume I Methodology Definition, SPC-91122-MC, Dec. 1991.

7. Gross, David C. and Stuckey, Lynn D. "Is Object-Oriented Design Sound Simulator Software Engineering?" 14th I/ITSEC Proceedings, Nov. 1992.

8. Rich McCabe, personal communication, April 1993.

9. Boehm, Barry W. and Scherlis, William L. "Megaprogramming (Preliminary Version)". STARS '92 Proceedings, Dec. 1992,

A COMMERCIAL ALTERNATIVE TO TACTICAL EQUIPMENT

Richard W. Pecile
Scientific-Atlanta, Inc.
San Diego, Ca.

Eddie Smith
Naval Training Systems Center
Orlando, Fl.

ABSTRACT

The solution to a training requirement is often implemented through a trade-off between the procurement of tactical equipment or a simulator. The choice usually depends on the differences between the high recurring costs of the tactical equipment versus the high non-recurring costs of a simulator. Simulators may also require the acceptance of compromises in training effectiveness.

A third alternative may satisfy the cost and performance requirements without compromise. This alternative uses the military specification tactical equipment designed and built to commercial standards. The relaxation of tactical environmental requirements for the benign classroom environment allows for the use of a commercial grade system. The key is to develop a system which is a functional equivalent of the tactical system. This is accomplished through use of commercial parts and components, resulting in an overall cost reduction.

Using a real world example based on the AN/BQR-22A, EC-15 Sonar Receiving Set, this paper traces specification and performance considerations, design strategies to shorten development schedules, and manufacturing approaches to minimize training and life cycle costs.

ABOUT THE AUTHORS

Richard Pecile is a senior systems engineer at Scientific-Atlanta's Signal Processing Systems division. He was the project engineer for the AN/BQR-22A, EC-15 Multi-Beam Processor test and integration effort. He is currently heading the design and development effort for the company's next generation parallel digital signal processing engine. Prior to joining Scientific-Atlanta in 1992, Mr. Pecile was the hardware design manager for the P-3A/B Tactics Trainer Device 2F142(T). Mr. Pecile holds an MES in Computer Science from Loyola College, Baltimore, Md. and a BS in Electrical Engineering Technology from the Pennsylvania State University at Harrisburg.

Eddie Smith is the Project Manager, Engineering (PJME) for Device 21H16, the Master Level Sonar Analysis Trainer, at the Naval Training Systems Center (NTSC), Orlando, Fl. Mr. Smith is presently in the Submarine Systems Branch and has extensive experience working with surface ship trainers. He has worked as a Reliability and Maintainability engineer at NTSC. He has a B.S. in Ocean Engineering from Florida Atlantic University and graduate courses from the University of Southern California in Systems Management and from the University of Central Florida in Engineering (Computer Science). His government employment career includes the Fleet Analysis Center, Corona, Ca., Navy Metrology Engineering Center, Pomona, Ca., Long Beach Shipyard, Ca., Naval Coastal Systems Center, Panama City, Fl., and the Naval Training Systems Center, Orlando, Fl.

A COMMERCIAL ALTERNATIVE TO TACTICAL EQUIPMENT

Richard W. Pecile
Scientific-Atlanta, Inc.
San Diego, Ca.

Eddie Smith
Naval Training Systems Center
Orlando, Fl.

INTRODUCTION

Shrinking defense budgets and accelerated procurement cycles have become the doctrine of the 90s. Government acquisition commands must seek new, innovative, and alternative means to acquire goods and services within, oftentimes, severely limited time and budget constraints while maintaining a superior armed forces. Defense contractors are similarly challenged with increased competition for fewer contracts, increasingly complex technical requirements, and challenging cost and schedule goals. Given this environment, alternative solutions to traditional military simulation and training systems must be examined and pursued.

Military training aims may be accomplished through stimulation of full MIL-Specification compliant tactical equipment, development of a simulation system, or a variant of these approaches. Development of simulation systems is always expensive, as is development of "bullet proof" tactical equipment. Tactical equipment is necessarily expensive because compromises should not be made on mission or safety critical items, such as performance, reliability, or maintainability. However, when that equipment will be used in a relatively benign environment such as a classroom, many cost drivers can be eliminated. Some are just unnecessary and others are mitigated by the fact that a failure is inconvenient, not life or mission threatening.

The choice between these two options typically depends upon the difference between the high recurring costs of the tactical equipment versus the high non-recurring costs of a simulator. While the non-recurring costs of tactical equipment may be low, the on-going costs are often quite high. This equipment is often difficult to obtain, impacting trainer development schedules. The use of tactical equipment which meets military specifications in a trainer system offers the advantage of being low risk because the development is complete and there will be no compromises in performance. The advantage of using simulators is that recurring costs are lower. However, the downside is that the cost of development is typically much higher. The use of simulators may also compromise training quality or performance.

The use of Commercial-Off-The-Shelf (COTS) or Non-Developmental Item (NDI) systems or subsystems is a method of trainer design preferred by the Naval Training System Center (NTSC) and other government acquisition agencies. In continuing this trend, the military training industry should evaluate commercialized tactical equipment for use in training. This path will bring together tactical equipment functionality and may significantly lower costs. Commercial components, while not MIL-STD, do meet industry quality standards and are driven by an unforgiving marketplace. An inferior product soon fails in the commercial sector. For trainer/training goals, the government can reasonably use the commercial marketplace to perform the function of its' extensive testing and qualification programs.

The environmental, reliability, and maintainability requirements for a training system are much less stringent than for tactical equipment. There exists a wide range of possible savings depending on engineering astuteness.

Using a real world example based on NTSC training Device 21H16, this paper traces specification and performance considerations, strategies for design to

shorten development schedules, and manufacturing approaches to reduce recurring costs. This design effort has resulted in a 70% cost savings in the included "tactical equipment" and has been accomplished in less than six months.

POSSIBLE APPROACHES

Many classroom/remote training systems require the inclusion of either real or simulated tactical equipments to increase realism or operational familiarity. Once a training requirement such as this is identified, one of the following three approaches can be used to address it:

1) Tactical Equipment

One possibility is to include the tactical equipment that is the focus of the training. The use of this equipment is often quite costly since tactical equipment is usually highly ruggedized and designed to operate under extreme conditions. As a result, the equipment must be built using highly reliable, rugged components that are usually significantly more expensive than their commercial equivalents. While the reliability requirements are essential in a potentially hostile environment, the training environment precludes the need for such stringent requirements and creates an opportunity for a less expensive approach.

2) Simulators

Simulators can provide a less expensive, effective alternative. Simulators usually need not meet the stringent reliability and maintainability requirements that are cost drivers in tactical equipment. Cost reduction is a goal of simulator design. Unfortunately, there are often trade-offs; simulators may not accurately emulate the tactical equipment, they can require a substantial design effort reducing the total cost savings, and there is always a risk that the design effort will be unsuccessful.

3) Conversion of Tactical Equipment

Conversion of full MIL-STD equipment to a "commercial grade" can produce a large cost savings and the least performance compromise. Many high cost contributors included in tactical systems can be eliminated or replaced in a training system. Expensive military components can be replaced with their commercial equivalents without affecting system performance.

Another benefit of this approach is that changes to operator interfaces can be minimized. Ideally, the conversion should be transparent to the operator. When accomplished, the training is more effective since the operator is trained on equipment that exactly emulates the tactical gear. This eases the transition from the classroom to the operational environment and reduces or eliminates time that would be required to train students how to use the simulator.

THE AN/BQR-22A, EC-15 EXAMPLE

The conversion of tactical equipment approach was used to meet a Naval Training Systems Center (NTSC) requirement. Ten multi-channel spectrum analyzer systems were required for the Master Level Sonar Analyst (MLSAT) Device 21H16's ten student stations. Device 21H16 is used to support the actual analysis training functions of the Master Level Sonar Analysis course being taught at the U.S. Naval Submarine School in New London, Connecticut.

An initial consideration, after determining the cost of tactical equipment, was to use a general purpose, commercially available, multi-channel spectrum analyzer system that would approximate the functions of the AN/BQR-22A, EC-15 Sonar Receiving Set currently being installed on all U.S. submarines. A commercial system was considered because sufficient funding was not available to purchase the actual shipboard militarized equipment.

The ideal approach to master level training would be to use actual shipboard equipment to teach detailed sonar analysis skills. Use of identified commercial alternative systems would not fulfill the desired training needs at this skill level. Therefore, NTSC and Scientific-Atlanta, the AN/BQR-22A, EC-15 designer/manufacturer, worked closely to develop a strategy which would provide the actual processing, display, and operator interface associated with the AN/BQR-22A, EC-15 sonar processor at a cost more in line with the available funding for Device 21H16. This was achieved by essentially commercializing the hardware in the AN/BQR-22A, EC-15 without changing the system's operational software. The result is a system, designated the SPS-900, which is operationally

identical to the AN/BQR-22A, EC-15 at 30% of the tactical system's initial purchase cost.

THE AN/BQR-22A, EC-15

The AN/BQR-22A, EC-15 Sonar system is a very high speed acoustic signal processing system used by the sonar operators to perform detailed sonar target detection and analysis. The AN/BQR-22A, EC-15 has eight analog inputs and receives inputs from the ship's spherical and towed arrays as well as the auxiliary intercept receiver, ship's noise monitoring hydrophones and omni-directional hydrophones through a sonar patch panel. The analysis functions performed by the AN/BQR-22A, EC-15 include narrowband, broadband, and DEMON. The time series instant replay and disk storage and playback capabilities of the AN/BQR-22A, EC-15 make it a uniquely flexible tool for transient analysis and event reconstruction.

The AN/BQR-22A, EC-15 consists of a processor unit and a display/control unit. These units can be either co-located or separated by as much as 300 feet. The processor unit consists of a time domain processor, frequency domain processor, and a power distribution unit. The display and control unit contains two color CRT monitors and the system control processor which includes the system keyboard and two 44 MByte bernoulli disk drives.

The system contains thirty-five circuit card assemblies (CCAs). The circuit cards are MIL-Spec, VMEbus based CCAs. Several of these CCAs utilize surface mount components. The color monitors display the acoustic information in several operator selectable formats as well as Processing Channel annotation and system status information. The AN/BQR-22A, EC-15 has full diagnostic and maintenance software which supports fault isolation to the board level. System control is accomplished using dedicated keyboard and pop-up menus. The control software resides on the system's bernoulli disk drives and can be easily updated by issuing a new software disk. A block diagram of the system is shown in Figure 1. Figure 2 shows a photograph of the system's processor and control/display units.

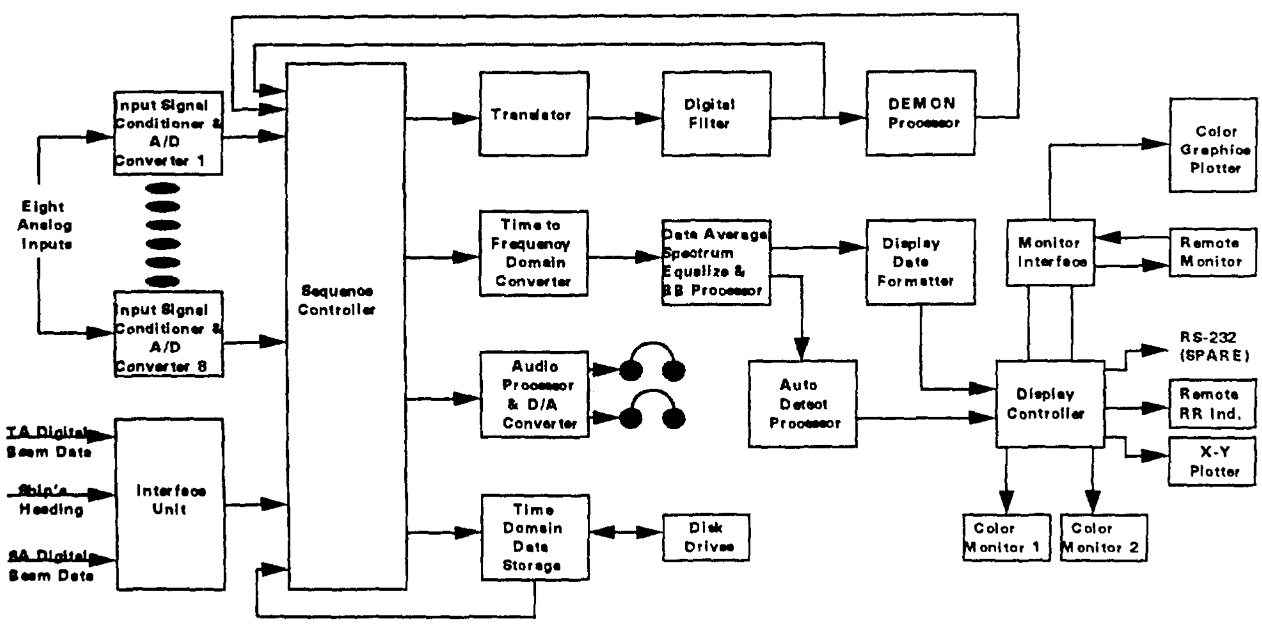


Figure 1
AN/BQR-22A EC-15
Functional Block Diagram

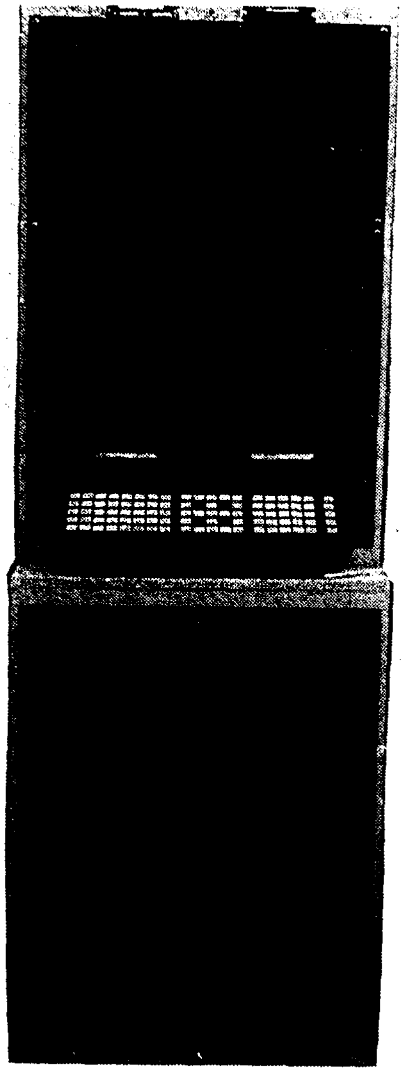


Figure 2
AN/BQR-22A, EC-15 System

The reliability and maintainability requirements imposed on the AN/BQR-22A, EC-15 are rigorous since it is ship's sonar equipment. Military standards for EMI, airbornre noise, structureborne noise, shock, vibration, and humidity are also imposed. These requirements were extreme for the target trainer system. Relaxation of environmental requirements during trainer system specification provides an opportunity to apply cost reduction techniques such as commercialization of tactical equipment.

SPECIFICATION CONSIDERATIONS

As is true of electronic systems in general, establishment of the SPS-900's system level requirements was key to the program's success. Using the existing AN/BQR-22A, EC-15 system specification as a basis, a specification delineating the required performance characteristics, operating environment, test criteria, reliability and maintainability requirements, etc. was created and agreed to by NTSC and Scientific-Atlanta.

At the outset, it was known that the SPS-900 would utilize commercial grade electronic components and that relaxation of environmental requirements was both necessary and desirable. In particular, the system's operating temperature range was reduced to 60° to 80°F and relative humidity range requirements were reduced to 50% to 75% non-condensing; ranges more appropriate for the anticipated components and within parameters required of the training device. Reduction or elimination of non-critical, cost driving requirements such as EMI/EMC levels, shock, vibration, noise, humidity, fungus proofing, etc. were also implemented in the system specification. Logistical requirements such as MTBF and MTTR were adjusted to reflect commercial grade components.

Actual system performance and functionality requirements were not changed from the tactical system specification in keeping with the conversion philosophy and goals. While this may be taken as a general goal of tactical equipment conversion projects, cost impacts versus associated training benefits should be carefully assessed during the specification phase.

Cooperation and interaction between Scientific Atlanta and NTSC during system specification development were crucial to the programs success. As an added benefit of the tactical equipment conversion approach to training, the system specification phase was accelerated by the existence of the tactical system specification for use as a template.

THE SPS-900 DESIGN

The SPS-900 performs all of the AN/BQR-22A, EC-15's operational functions with an identical user interface. It is designed and manufactured to best commercial practices. Whenever practical, the AN/BQR-

22A, EC-15's military components were replaced by their commercial equivalent. For example, the highly ruggedized color CRT monitors, each costing over $6000 were replaced with commercial computer monitors costing approximately $1200 each. Similar cost savings were realized in system power supplies, cables, and the power distribution unit.

The first step in the actual design process was determination of material costs for existing AN/BQR-22A, EC-15 assemblies and subassemblies. Actual costs can typically be captured from on-line Management Information System data bases containing purchase order histories and component cost breakdowns. Examination of this information revealed the highest costing assemblies with the greatest potential for cost reduction through commercialization. Additionally, labor charging activities were captured to determine which, if any, manufacturing processes could be modified or eliminated as a result of commercialization.

A task force consisting of engineering, manufacturing, test, purchasing, and quality personnel was established to implement the actual design changes. Recommendations for replacement commercial components were examined by task force members to insure that an overall cost reduction would be realized as a result of the change. Savings of 50% in the cost of power supply may be of no benefit if a new test fixture must be designed or if the procurement lead time is excessive, for example.

Particular attention was given to the conversion of circuit cards given the large number of complex cards contained in the system. The SPS-900 uses the circuit card assemblies contained in the AN/BQR-22A, EC-15. However, they were modified to reduce costs. Every military grade microcircuit was analyzed to determine the performance, cost and availability of commercial grade equivalents. This research task was easily performed by an engineering aide using published cross reference information, with a more senior engineer providing guidance and approving choices. The effort to commercialize CCAs was accomplished in less than one month. Since circuit card layouts were unchanged, existing automated manufacturing tools, machine programming tapes, etc. were usable without change. No effort was made to redesign circuits to accommodate commercial components. Components were specified which generally matched the existing devices' electrical and physical parameters. Where this was not possible, engineering analyzed the impact on circuit performance before approving the recommendation. Likewise, no effort was made to redesign circuit cards to condense card count, improve performance, etc. as this would increase non-recurring design and documentation costs.

Significant material cost savings were realized by replacing military grade integrated circuits with commercial equivalents. Figure 3 shows a cost comparison of some commonly used integrated circuits. While not shown, MIL-M-38510 QPL parts are often more costly than MIL-STD-883 grade components. The Department of Defense recognizes the cost savings potential of using high reliability commercial grade integrated circuits in military systems and is presently drafting new policies governing their use.[1]

Generic Part No.	MIL-STD-883 Part Cost	Commercial Equivalent Cost
GAL22V10-20	$ 37.30	$ 5.77
54FCT373	$ 6.24	$ 0.75
68000-10	$122.66	$31.40
TL072	$ 3.31	$ 2.00

Figure 3
Typical Military vs. Commercial Grade Integrated Circuit Cost Comparison

The SPS-900 software is identical to that used on the AN/BQR-22A, EC-15. This represents a significant advantage of the converted tactical equipment approach to training over custom designed simulators or inclusion of militarized equipments. Software development cost and risk are eliminated since the operational code has been developed, documented, and thoroughly tested. Also, software modifications to the tactical equipment are directly incorporated in the training device, ensuring trainer capabilities are kept current.

Manufacturing processes were examined to determine whether labor savings could be realized as a result of the commercialization efforts. Since the AN/BQR-22A, EC-15 manufacture is highly automated, especially circuit card production, every effort was made to retain the existing process flow. For lower volume, labor intensive processes however, the manufacturing

process should be closely examined for cost savings. For example, requirements for weapons specification soldering may be relaxed for trainer programs. Cost driving manufacturing requirements should be flagged during contractual negotiations.

Significant savings were also realized in the SPS-900 program test phases, both initial acceptance testing and production testing. At the circuit board level, test fixtures and automatic test programs were directly reusable from the AN/BQR-22A, EC-15 program since the only changes to these circuit cards were the inclusion of commercial grade components. At the system test level, existing procedures and documents required only minor modifications to account for specific physical differences from the original militarized system, again resulting in cost savings.

The SPS-900's technical documentation package was another area which benefited from this approach. Existing technical manuals required only slight modifications to account for differences in component part numbers or physical characteristics. Existing, high quality software documentation was usable without change.

LOGISTICS CONSIDERATIONS

Life cycle costs and supportability are important considerations for military equipments, particularly low production volume systems such as trainers. Strategies used in design of COTS/NDI based systems to insure affordable, supportable systems are equally applicable to converted tactical systems. Contractors should look for availability of second sources, guaranteed product migration paths, and vendor support commitments in the commercial component selection process. Converted tactical systems have a an additional source of supply for obsolete commercial components; the equivalent military component may be obtainable from the supply system for direct substitution.

SPS-900 PROGRAM RESULTS

A firm fixed price, sole source contract for ten SPS-900 systems to form Device 21H16's student stations was issued on 10 March 92. Delivery of the first SPS-900 system occurred 8 months after contract award at 30% of the AN/BQR-22A, EC-15's cost. All ten systems have been delivered and are operational at the Naval Submarine School, New London, Connecticut. A photograph of Device 21H16 is shown in Figure 4. As can be seen, the SPS-900 (right hand bay of each student station) is essentially identical to the AN/BQR-22A, EC-15 shown in Figure 2 with the exception of the tilt angle of the upper monitor, done for trainer ergonometric reasons.

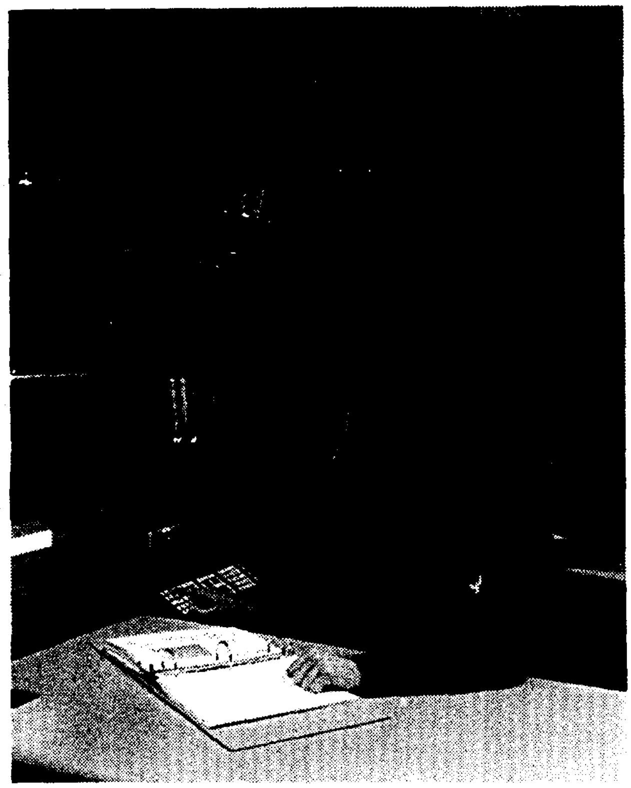


Figure 4
SPS-900 System

Faced with a diminished Soviet military threat and shrinking defense budgets, Scientific-Atlanta, like many other defense contractors, is seeking ways to convert its military products and expertise to meet the needs and demands of a commercial marketplace. Lessons learned in the SPS-900 program have been applied to yet another evolutionary product, the SPS-1000. This high speed, multi-channel signal analyzer meets the needs of research laboratories and universities. Simple SPS-900 hardware and software modifications to eliminate unnecessary features, provide added features, and insure declassification requirements resulted in a system with world-wide market potential.

SUMMARY/CONCLUSIONS

The decision to base MLSAT Device 21H16 on converted operational equipment was made after thorough performance and cost trade-off analyses of candidate approaches. Clearly, this approach is not applicable to all training devices. However, discounting desk top computer emulations, tactical systems with imbedded training capabilities, on-board trainers, dock side trainers, and the like, a large class of trainers/simulators which utilize tactical equipment in the loop or which simulate tactical equipments still remains. For this class training device, careful consideration should be given to the option described in this paper, i.e. commercializing included tactical systems.

Trainer systems planning to include tactical equipments may realize significant cost savings from this approach. Some potential benefits include:

- Lower system recurring cost.
- Lower on-going maintenance costs.
- Shortened procurement cycles.
- Improved spare parts supply

Conversely, trainer systems planning to simulate tactical equipments may also benefit from this approach through:

- Reduced non-recurring design costs.
- Shortened development cycles.
- Exacting emulation of tactical equipment functions and operator-machine interfaces.
- Reduced documentation costs.
- Reduced development schedule risks.

The suggested approach was proven in the very successful design of acoustic analysis training Device 21H16. Assuming availability of adequate design documentation, the conversion of tactical equipment to commercial grade equivalents for training purposes is a viable, cost effective solution for a large class of trainer systems.

REFERENCES

1. Keller, J. "DoD Shift to Commercial ICs Worries Industry", Military & Aerospace Electronics, Vol. 4, No. 6, June 1993.

APPENDED TANK FULL-CREW INTERACTIVE SIMULATOR TRAINERS

Richard D. Gillem, Richard D. Leavitt, and Daniel E. Yuchnovicz
Computer Sciences Corporation, Integrated Systems Division, Training Systems Center
Newport News, Virginia, and Orlando, Florida

ABSTRACT

This paper discusses the benefits of appended full-crew training systems for armored fighting vehicles. It shows the benefits appended trainers have over institutional trainers. It describes two similar yet different appended trainer designs and the design challenges involved in the development of each. The paper also discusses the employment experience of an existing appended trainer and the resulting benefits. While this technology is applicable to any armored fighting vehicle, this paper will address only tank training systems.

The successful pioneering of the development of appended full-crew, interactive tank gunnery skills training devices has occurred. Devices were delivered to the U.S. Marine Corps (USMC) Reserve for the M60A1 tank in 1989 and for the M1A1 tank in 1993, and demands for application to other fighting vehicles are increasing. An M60A1 trainer was used by USMC Reserve units aboard the USS Tarawa to train tank crews en route to DESERT STORM.

For optimum effectiveness, appended trainers must be capable of training the entire tank crew -- tank commander, gunner, driver, and loader -- as a single integral crew. They must have high fidelity and meet all of the requirements for a precision gunnery trainer. They must literally turn the entire tank into a simulator. Since the trainers are deployable, tank crews can maintain gunnery skill proficiency wherever they are deployed.

ABOUT THE AUTHORS

Richard D. Gillem is the Program Manager for Full-crew Interactive Simulator Trainer (FIST) operations for Computer Sciences Corporation's Training Systems Center. He was the Program Manager for both the M60A1 and M1A1 MCTFIST systems. He is a retired Army Colonel with more than 30 years military training experience. He holds a B.S. in Industrial Management from Georgia Tech and an M.S. from Shippensburg State University.

Richard D. Leavitt is the NTSC/STRICOM interface and the Orlando, FL, Office Manager for Computer Sciences Corporation's Training Systems Center. He is a Retired USMC Master Gunnery Sergeant with more than 14 years experience in the design, development, and fielding of military training systems and devices. He has been involved in the MCTFIST program since its inception in 1937. He holds a B.B.A. from Orlando College and an A.S. from Northern Virginia Community College.

Daniel E. Yuchnovicz is a Senior Project Engineer for Computer Sciences Corporation's Training Systems Center. He has more than 12 years experience in system design and integration. He was the principal engineer on the M60A1 and M1A1 MCTFIST systems. He holds a B.S. in Electrical Engineering from Western Michigan University and is pursuing an M.S. in Computer Science.

APPENDED TANK FULL-CREW INTERACTIVE SIMULATOR TRAINERS

Richard D. Gill[illegible], Richard D. Leavitt, and Daniel E. Yuchnovicz

INTRODUCTION

On today's lethal and fast-moving armored battlefield, a tank crew's proficiency in combat gunnery skills is often the decisive factor in the outcome of any engagement. Tank crew proficiency is only possible if the crew members are well trained in all aspects of their tank's operation and capabilities in a tactical environment. Developing and sustaining crew gunnery skills at a level of readiness that permits rapid transition to combat represents a continuing challenge for all tank units.

Traditional tank crew training is very time-intensive and costly. It requires regular access to maneuver and firing ranges and is expensive in terms of ammunition, repair parts, fuel, and range support. The schedule and cost constraints associated with traditional tank crew training are particularly acute. As training costs escalate and environmental restrictions reduce training area options, a realistic, low-cost alternative approach to effective tank crew gunnery training is needed. A viable alternative is now possible.

The initial challenge was to determine if it was feasible to develop a realistic, low-cost appended simulator that could be effectively used to train entire tank crews in the actual vehicles in which they would be expected to fight. The concept was to append visual display monitors to the tank sight and vision apertures to provide the tank crew with a realistic view of a battlefield engagement. The projected visual images had to be integrated with the actions of the tank crew members to provide a realistic training environment. Finally, the entire system had to be driven by a personal computer if a low-cost solution was to be achieved.

The feasibility of this concept was demonstrated in 1986 when the prototype of a Full-crew Interactive Simulator Trainer (FIST) was developed by Computer Sciences Corporation (CSC). This prototype system drove the requirements for a National Guard FIST procurement (GUARDFIST I) and was used by the Marine Corps in developing the M60A1 Marine Corps Tank Full-crew Interactive Simulator Trainer (MCTFIST) system requirements.

Since the development of the first prototype, technology has taken major strides forward, especially in the area of image generation. Multiple videodisc players with standard resolution monitors were used for the first prototypes due to the cost and size of image generators at the time, and the lack of computer boards with high resolution video output. However, the current fourth-generation appended trainer now in use on M1A1 tanks uses smaller, less expensive, more capable image generators with high-resolution display capabilities.

MCTFIST DESCRIPTION

The MCTFIST family of trainers is the only full-crew appended training system that does not require some operation of the tank. It permits the full tank crew to develop and hone collective gunnery skills while training in a stationary powerless vehicle. Because MCTFIST makes use of the tank's electronic and mechanical systems, it literally converts the actual tank to a simulator. Therefore, it is about as close to an embedded training system as can be achieved with today's fleet of tanks.

MCTFIST requires no special facility or environmentally controlled area. A covered tank maintenance bay or other similar area is sufficient. It is easily transportable in organic packing cases. For example, the largest component of the M60A1 MCTFIST, the instructor/operator station (IOS), is approximately 2 feet by 2 feet by 3 feet and weighs about 150 pounds. For Marine Corps applications, MCTFIST can and has been used

aboard ship to train embarked tank units. Additionally, it has been used for remedial training during annual tank gunnery exercises in a desert environment. With outside air temperatures up to 116° F, the trainer was operated in a maintenance tent while powered by a standard USMC MEP-0005 generator.

Operation requires no special computer skills. The average tank commander (TC) can learn the Instructor/Operator (IO) procedures in one hour. With one day's training, a tank crew can install and calibrate the system in about an hour without special tools. Periodic operator maintenance is limited to keeping the system clean and extracting training records from the database onto floppy disks for future reference.

Users state that MCTFIST is especially effective in developing and sustaining TC and gunner proficiency in gunnery techniques and manipulation of controls, and in cross-training crew members in all crew duty positions. Dramatic improvements have been noted in crew communication, coordination, and the ability to quickly and accurately place "steel on target" after as little as one hour of training.

While MCTFIST has been developed and proven on the M60A1 and M1A1 tanks (see Figures 1 and 2), its baseline modular components are adaptable for use on any tracked or wheeled fighting vehicle anywhere in the world.

M60A1 MCTFIST

The M60A1 MCTFIST (see Figure 1) is a videodisc-based full-crew tank gunnery skills training system that brings battlefield realism to the training arena at a very low price. Because the system uses actual filmed background terrain, a variety of realistic training environments -- desert, winter, etc. -- can be used. The M60A1 MCTFIST tank crew can "drive" the tank into and out of hull- and turret-down positions behind actual terrain features. Crews can train on filmed scenery of the actual areas in which they expect to fight.

M60A1 MCTFIST OPERATION

The system consists of both off-the-shelf and custom-designed components. Video monitor assemblies are appended to the principal tank sights and vision apertures and display the visual effects of simulated engagements. Sensors placed on the key tank controls permit realistic simulation of the tank's movement and fire control systems.

The TC has an out-the-hatch view of the tactical situation. The TC uses the tank optical range finder, intercom, and TC weapons system controls to acquire targets, determine range, and fire the main gun and coaxial machine gun.

The gunner uses the tank primary sight, ballistic sight, and gunner controls to acquire targets, select ammunition, and fire the main gun and coaxial machine gun.

The driver observes terrain through his center vision block and uses the steering T-bar, gear selector, accelerator and brake pedals, and appended speed and RPM gauges to control the tank's simulated movement.

The loader selects and simulates loading main gun dummy rounds and sets the safe/fire switch for both the main gun and coaxial machine gun.

The training is managed from a compact IOS mounted in a shock-proof case for easy transportation. The IOS contains two videodisc players to display the actual filmed terrain. A microcomputer controls the simulation system, automatically scores engagements, and collects training management data. Sound equipment realistically replicates engine and weapons sounds. A high-speed printer provides hard copy training records of crew performance.

In the videodisc variant, the IO selects from among 45 gunnery tasks from the M60A1 tank combat tables, including precision, battlesight, and degraded gunnery engagements. Three types of stationary and moving targets can be engaged at various ranges using stabilized or unstabilized modes. The IO briefs the crew on exercise tasks, conditions, and standards over the intercom system. He monitors the ongoing tactical situation as seen by the TC, analyzes individual and crew performance from the data available on his display console, and conducts comprehensive after-action reviews with the crew members.

M60A1 MCTFIST CONFIGURATION

The M60A1 MCTFIST system is made up of six subsystems: data processing, video, optical audio, crew control sensors, and data acquisition. All of the components are commercial off-the-shelf (COTS) with the exception of the optical and sensor subsystems. The system features a modular design that allows subsystem components to be modified for easy adaptation to other armored fighting vehicles.

Data Processing Subsystem

All aspects of trainer operation and control are performed by an IBM AT-compatible single-board computer. Inputs to the processor come from the IO keyboard and data acquisition subsystems. The IO keyboard is used to initialize the trainer, select exercises, input data, and perform diagnostic/maintenance functions. Outputs from the processor are used to control the video and audio subsystems as well as to present the IO with a real-time exercise status display.

Video Subsystem

All images of background terrain, targets, and weapons effects are generated by the video subsystem. The composite video image is presented to the tank crew via the optical subsystem. The video subsystem is made up of dual videodisc players, four image processors, seven video monitors and video distribution amplifiers. At the heart of the video subsystem are the image processing boards that make this trainer a reality. Each frame from the videodisc is fed to the image processing boards, where apparent vehicle and turret motion, targets, and weapons effects are generated.

Apparent motion through the tactical terrain is achieved by stepping the videodisc player in forward or reverse in response to the driver's controls. Turret motion is simulated by panning and scrolling the video image in response to the gunner's and TC's controls. Target and weapon effects graphics are superimposed on the video frames at positions coordinated to the current terrain features. Processed images are then presented to the tank crew through monitors appended to the tank crew's primary sights and vision apertures. Four unique processed views are provided: the TC's out-the-hatch view; the gunner's primary and secondary gunsight views; the TC's coincidence range finder view; and the driver's center vision block view.

Exercise segments can cover several thousand feet of filmed terrain. Segments may be linked together into one exercise so the TC is faced with binary turning decisions, e.g., the driver can continue to move down the current path or can make a turn onto a new path at selected points to reach firing positions.

Optical Subsystem

The optical subsystem locates and presents a focused and collimated image to the tank crew's primary vision apertures. It consists of custom mechanical mounts, which locate video monitors at the vision apertures, and optics, which allow the tank's telescopic sights to focus on the monitor's face. Each mount has been designed to be installed by a minimally trained tank crew without the use of tools. The optical system allows the gunner to take full advantage of the actual gunsight reticles. Optical assemblies are not required for the out-the-hatch and driver views.

Audio Subsystem

An audio distribution system provides two-way communications between the IO and the crew. It also injects sound effects into the tank intercom system. The audio distribution system is tied directly into the tank intercom system. The IO is provided with a microphone/headset. An external speaker at the IOS allows observers to hear both the IO's comments and the crew's fire commands/conversations within the tank. The audio subsystem uses an audio digitizing board to present the tank crew with aural information from the engine, main gun, and coaxial machine gun. These sounds are actual recordings of the tank engine at various speeds and gear positions, and weapon sounds recorded on the live-fire range.

Data Acquisition Subsystem

Data is transferred to and from the tank over the data acquisition subsystem. Several times per second the processor reads the current position of each sensor and control switch. The data acquisition subsystem interfaces directly to the gunner's control and stabilization switch boxes, the gunner's control handle, and the loader's safe/fire switches. The tank's cable harnesses are removed from these connectors, and replaced by the simulator's data acquisition cables. Therefore, each crew member can use each sensed control or switch as it would normally be used without interference.

M60A1 MCTFIST SOFTWARE

The M60A1 MCTFIST software was designed in a modular fashion. The language used was C, with portions written in Assembler. A windowing package and database package were integrated into the system for the control of the IO input/output screens and for maintenance of the database.

A graphics subsystem was designed to handle the presentation of target images and visual weapons effects over background scenery. A database maintains crew performance data for each exercise for later review and for record. A real-time status screen presents ongoing exercise data and summary information for IO evaluation at task termination. The database contains information such as crew member names, the exercise, task, conditions, standards, ammunition used, time rounds were fired, and locations of hits and misses with respect to the kill zone of the target.

M60A1 MCTFIST DESIGN CHALLENGES

Appended trainers present unique challenges not encountered in institutional trainers. During the design process it was discovered that there are significant variances in the dimensions of M60A1 tank hulls and turrets that impact the placement of optical assemblies. Therefore, adjustable mounts were required for all appended fittings and supports. The most challenging obstacle was finding a design solution to simulate the optical range finder. A proprietary image "ghosting" technique was developed and an extra monitor with a collimating lens was added to simulate the optical range finder views.

M60A1 MCTFIST Goes To War

The two initial M60A1 simulators have had a very colorful history. At their annual active duty training in 1990, a USMC Reserve company's tank crews achieved 95 percent first-round hits on Table VII, the most difficult exercise fired. This resulted in saving two days of live-fire range time and over 300 rounds of main gun training ammo costing over $120,000.

In November 1990, Company A, 4th Tank Battalion, based at San Diego, CA, was mobilized and ordered to deploy to the Persian Gulf aboard amphibious warfare ships. The Company loaded MCTFIST, packed in its travel cases, onto a 2 1/2 ton truck which was positioned beside an M60A1 tank in the Well Deck of the USS Tarawa. This loading scheme permitted the conduct of initial training from the truck bed for filler personnel received shortly before deployment and sustainment training for more experienced crews. This type of shipboard training was a ***FIRST*** for armor forces of any country. Never before had an armor unit been able to conduct realistic and challenging full-crew gunnery skill training while in transit onboard a ship. Illustrating the simplicity and robustness of the M60A1 MCTFIST was its satisfactory functioning on 110 volts AC shipboard power and its operating in the highest sea state in which personnel were permitted in the well deck although not initially designed for shipboard operation. Another demonstration of system versatility was that it was installed on an M60A1 tank equipped with reactive armor tiles that materially altered the tank geometry from the design baseline.

Upon their return from DESERT STORM, the USMC Reserve TCs credited their M60A1 MCTFIST training with producing a high level of crew proficiency and confidence prior to entering combat for the first time.

M1A1 MCTFIST

The M60A1 MCTFIST performance during DESERT STORM resulted in an additional contract to CSC to upgrade the M60A1 from a videodisc-based system to an image generator-based M1A1 MCTFIST. The M1A1 MCTFIST is now in service with the Marines at the Marine Corps Reserve Center at Yakima, WA. What follows is a design overview and discussion of the challenges faced during the design and development of the M1A1 MCTFIST.

M1A1 MCTFIST Design Overview

The M1A1 MCTFIST system functional flow diagram illustrates the relationships between trainer subsystems and the tank/crew stations. M1A1 MCTFIST comprises seven subsystems which interface with the M1A1 optical, audio, electrical, and mechanical systems (see Figure 3). They are:

- o Simulation and Control
- o Data Acquisition
- o Graphics
- o Video
- o Optical
- o Audio
- o Power Control

Simulation and Control. All phases of system operation, pre-test, in-test, post-test, and diagnostic operation are controlled by the Simulation and Control subsystem. This subsystem resides in a compact IOS. All of the software required for scenario selection, execution, and analysis operate on a special purpose 80486 processor. The M1A1 MCTFIST IO controls the simulator from the IOS via a point and click menuing system. The menuing system guides the IO through scenario development using on-screen prompts, instructions, and on-line help.

During scenario execution, the IO monitors the crew's progress from the IOS via two monitors. One monitor shows the live video as seen through the gunner's primary sight (GPS), GPS extension (GPSE), and gunner's auxiliary sight (GAS). A second monitor shows either real-time data gathered as the scenario progresses, or the live video seen through the TC's forward vision block and the driver's forward vision block. The IO can select the real-time or video displays on the second monitor at will. The real-time display graphically shows the current settings of all crew controls and the headings of the tank hull, turret, and commander's weapon station (CWS). When a target is engaged, the reticle aim error from the target's center of mass is instantly displayed to the IO. During an exercise the IO can freeze, thaw, stop, or restart the current exercise. All statistics gathered during the scenario are presented to the IO after the crew finishes the engagement. The IO can then finish the semi-automated scoring process and add free-form comments if necessary. Hard-copy reports are available if desired by the IO. The M1A1 MCTFIST contains an Automated Training Management Information (ATMI) system. This useful feature enables the IO to perform an extensive analyses of exercise performance data and generate reports for training managers. The reports show trends and allow crew cross-training analysis, and are used to support crew composition decisions.

Data Acquisition. The M1A1 MCTFIST data acquisition system acquires crew control movement from the tank crew stations. If any of the crew controls are disconnected from the tank's turret network box (TNB) and are interfaced directly to the MCTFIST data acquisition system. Settings of mechanical controls such as the driver's brake are acquired via appended electromechanical sensors. Currently the data acquisition system acquires about 70 lines of digital discrete data representing switch settings and eight channels of analog data representing turret and hull movement controls. The data acquisition system also provides digital relay control of 23 crew station status indicator lamps and two analog channels for RPM and MPH gauges. A sensor interface box (SIB) is located within the tank turret to acquire over 200 signals from the four crew stations via five short wiring harnesses. The SIB acts as a signal routing matrix which reduces the 200 tank signals to the digital and analog signals acquired by the data acquisition system.

A special feature of the M1A1 MCTFIST is that the connectors on the data acquisition harnesses can be verified for proper connection

via software. This is accomplished by using the loop-back provisions provided in every connector in the M1A1 tank. Loopbacks are checked at scenario start-up, and, if an error is found, the system automatically changes to the diagnostic state. The IO is then presented with a real-time display that shows the offending loopback as well as the real-time status of all digital and analog channels.

Graphics. M1A1 MCTFIST graphics are rendered by an image generator (IG). The IG renders fully anti-aliased and phototextured images at a 30 hertz (Hz) update rate. Current system capacities include 120,000 polygons per second and 152 million pixels per second divided over three video output buffers (channels). One channel is split-screened, with the upper half presenting the TC's forward vision block display and the lower half presenting the driver's vision block display. A split-screen view makes efficient use of the video channel in that it matches the horizontal aspect of the vision blocks. The TC's and driver's views are computed and displayed at 40 degrees horizontal by 12 degrees vertical. The second and third channels are devoted to rendering the gun sight views (GPS, GPSE and GAS). Light entering the GPS and GPSE originates at a shared, common point, enabling one monitor and one IG channel to satisfy the display requirements of both sights. Imagery from the same channel is also presented to the GAS. Imagery for the GPSE is computed and displayed at either 6.3 degrees circular or 22 degrees circular field-of-view (FOV), depending upon the magnification switch setting sensed at the gunner's station. Imagery for the GAS is computed and displayed at C degrees circular FOV when the infrared electro-optical sensor detects that the gunner is looking through the GAS. A unique feature of the M1A1 MCTFIST gunsight is that the imagery in the central 35 percent of the FOV is rendered with the third channel and is presented at a resolution that allows identification of targets at ranges more than twice that of any other M1A1 simulator. This allows M1A1 MCTFIST to take full advantage of the capabilities of the M1A1 optical system.

Video. The M1A1 MCTFIST video subsystem interfaces the IG to the M1A1 tank optics via the optics subsystem. Each channel output from the IG is fed into video distribution amplifiers (DAs). The outputs from the video DAs are fed to eight identical monitors which are appended to the tank and positioned atop the IOS. All monitors are operated at a rate of 768 pixels by 768 lines. One video switch is used to display either the split-screened TC/driver video or the real-time status video on one IOS monitor. A second video switch is used to turn off the imagery displayed on the GAS video monitor when simulating a turret defilade position. The GAS entrance aperture is positioned about one meter lower on the turret than the GPSE entrance aperture. Using the video switch to block out the lower GAS view when in a defilade position allows the use of a single IG channel to simulate the views from the GAS and GPS.

Optical. The M1A1 MCTFIST optical subsystem is made up of two types of optical assemblies. The first type provides direct view displays for the TC and driver vision blocks. A second type provides collimated displays for the gunsights (GPS, GPSE, and GAS).

The direct view optical assemblies consist of a monitor, mechanical mount, and external light limiting shields. The monitors are 1024 pixels by 768 lines in resolution. They are held in front of the vision blocks on the turret and hull by mechanical mounts that are unique to each vision block. The driver views the driver imagery directly through the vision block. Folding optics are incorporated into the TC's mechanical mount in order to position the TC's image in the TC's vision block. The mounts provide secure positioning of the monitors and strain relief for the video cables. External light-limiting material is used in the form of heavy black cloth that is held to the mount by Velcro, and to the tank with magnets that are stitched into the cloth.

The collimated displays are of a proprietary design developed for this application. Two collimated optical assemblies are used for the gunsights, one for the GPSE and an identical one for the GAS. Each optical assembly positions two monitors in front of each gunsight. One monitor is collimated into the gunsight and is referred to as the background monitor. The second monitor displays the high-resolution inset image. This high-resolution

image is directed into the central portion of the FOV by an optical system contained within the optical assembly. The collimating mounts also provide the mechanical features of secure monitor mounting to the tank hull and turret with video cable strain relief.

Audio. All audio cues are generated in real time in response to crew actions. A sample player is used to play back actual digitized sounds of various tank sounds. All generated audio sounds are injected into the crew intercom system along with the IO voice channel and are heard by a crew member via his helmet. The IO can control the volume of his microphone, headset, and simulated tank operational sounds via an audio mixer located at the IOS. Control of the sample player is by the Musical Instrument Digital Interface (MIDI) in the Simulation and Control Subsystem. Simulated sounds include the turbine engine, track clatter, turret gear whine, main gun report, machinegun report, ammo door and breech open/close, and shell basecase discard inside the turret.

Power Control. All power to the system is controlled via a passkey located on the IOS. Once the passkey is activated, the IO may operate the IOS in a stand-alone mode for records analysis with the ATMI or in the on-line mode for simulator operation. Power conditioning (regulation and surge suppression) is provided for the 115/208 power circuits. The system consumes only 2800 watts during operation.

Design Challenges. Solving The Identification Friend or Foe (IFF) problem was probably the greatest challenge. The M1A1 MCTFIST is the first appended or any other type of tank trainer that allows the TC and gunner to distinguish between friendly and enemy targets at ranges well in excess of 2000 meters. Lessons learned from the Gulf War reinforced the need to provide tank gunnery trainers that can provide IFF situations at realistic engagement distances. Other tank gunnery trainers may reinforce the notion that if it looks like a tank beyond 1000 meters that it must be a hostile target. This is due to limitations of their visual display system, i.e., detail seen in the target. Most gunnery trainers provide around 800 to 1000 lines of vertical resolution in the gunsight FOV using a single, high-resolution (1024 lines) monitor. This will allow a gunner to perform IFF tasks out to ranges of about 1200 meters.

As described earlier, M1A1 MCTFIST uses a dual-monitor approach to the gunsight visual display. The primary gunsight provides the gunner with a 6.3 degree FOV at a 10 power magnification. The image for this 6.3 degree FOV is displayed on one monitor that is collimated into the gunsight and is referred to as the background or direct view monitor. The system uses a 768-line resolution monitor which normally would limit positive target identification to a range of only 900 meters. To improve the range at which IFF tasks can be performed, higher resolution monitors must be used in order to provide more lines of resolution in the 6.3 degree FOV. Monitors that exceed 1024 lines are custom-made and expensive, or are too large to append to the tank as would be the case for a High Definition Television Monitor (HDTV). HDTV monitors will provide only about 1200 lines of resolution and would not increase the IFF ranges substantially. To increase the resolution in the M1A1 MCTFIST, the second monitor displays the central 1.8 degrees of the 6.3 degree FOV, but at a much higher, 30 power magnification. All 768 lines of the second monitor are used to draw the 1.8 degree central image. The key to super-high resolution then is that the 1.8 degree central image is compressed into the center of the 6.3 degree background image, i.e., a 1.8 degree central circle of the background is replaced by the high-resolution inset. This inset provides the resolution equivalent of a 2600-line monitor and more than doubles the range at which target identification can be made. A proprietary method is used to smoothly transition the central image with the surrounding image.

Sizing of the central high-resolution inset was chosen to completely surround the 1.2 degree wide gunsight reticle. Therefore, the reticle is rendered at three times normal size and shrunk down into the center of the gunsight FOV. This provides the gunner with a reticle that is as discreet as that found in the tank, and is sharper than a reticle rendered on a 1024 line monitor. With this "high-resolution inset," the gunner and TC can perform IFF tasks out to

and beyond the engagement ranges required in the tank combat gunnery tables.

Developing a user-friendly system was another challenge. The M1A1 MCTFIST can be appended to the tank by a trained tank crew in just over an hour. Accomplishing this task required considerable attention to human factors engineering in five areas: user documentation, cable harnesses, appended sensors, appended optical assemblies, and video cable harnesses.

The M1A1 MCTFIST user's manual was developed along the lines of a technical manual format that the tank crews are familiar with. Text was minimized and is used to support accompanying drawings. The system manual is also provided in computer form as a computer-based training (CBT) product that can be run on the IOS or any PC.

Many of the data acquisitions points within the turret and hull are electrical connections to actual tank equipment. In some cases electro-mechanical sensors are appended to the component being sensed, e.g., hydraulic service brake motion. None of the eight appended sensors take longer than two minutes to install, and a 30-second installation time is more typical. Speedy installation was accomplished by using sensors that have either magnetic bases or simple clamp-on mounting brackets. No special tools are required.

Tank Variations. Several models of the Abrams M1A1 Main Battle Tank exist. The basic M1A1 tank was upgraded to the M1A1 Common, which is more heavily armored. The outer dimensions of some components are noticeably different between the two models. Other variations between tanks occur in the GPS and GAS optics. The M1A1 MCTFIST design accommodates these variations. As in the M60A1 MCTFIST, every mount is designed to allow for the variations in tank hulls and armor.

SUMMARY

Since 1986 efforts have been under way to design low-cost armored vehicle trainers that can be appended to the actual vehicle, turning it into a complete full-crew training system that allows crew members to train the way they will fight, as an entire team (crew). The MCTFIST simulators are existing proof that appended full-crew trainers are not only feasible, but offer very realistic and effective training environments. Armored vehicle crew members can now use their full-crew trainers while in transit, while in staging areas prior to the initiation of combat operations, or while in a remote location under standard generator power.

Because of the modular design methodology utilized in these systems, with minimal modifications, additional armored vehicles can now utilize the same components, software, and databases, saving developmental time and money. As the Marines that used the M60A1 MCTFIST on board the USS Tarawa found out, a decisive edge can be achieved and maintained by taking trainers to war.

THIS

PAGE

IS

MISSING

IN

ORIGINAL

DOCUMENT

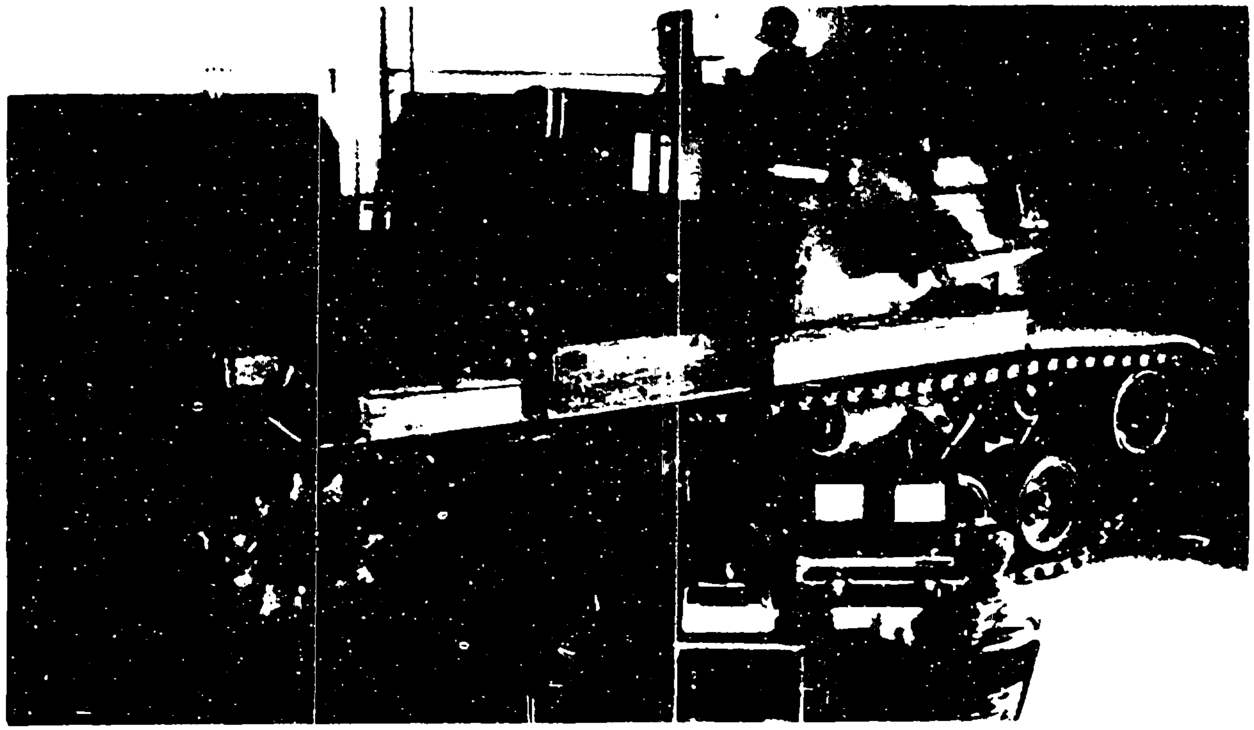


Figure 1. M60A1 MCTFIST

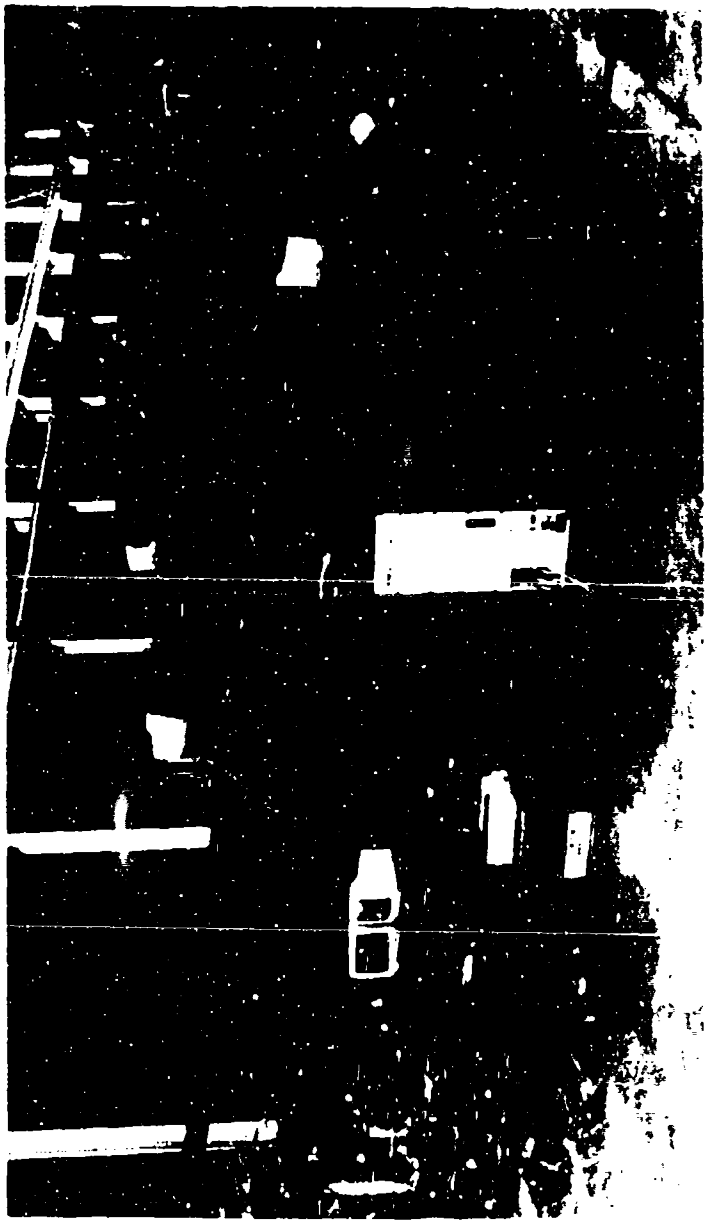


Figure 2. M1A1 MCTFIST

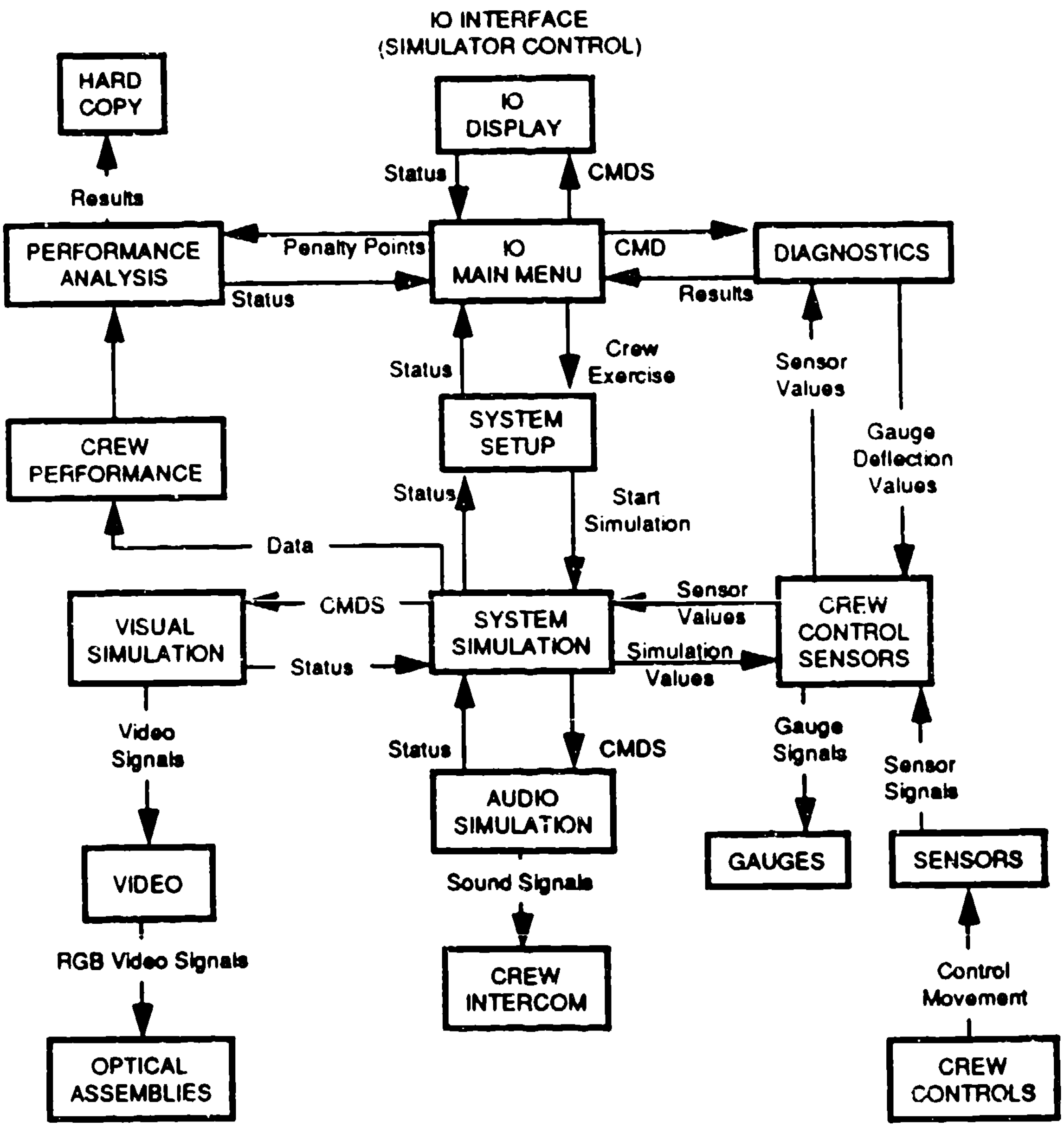


Figure 3. M1A1 MCTFIST Functional Flow

A COMPARISON OF TRUCK DRIVING INSTRUCTION USING SIMULATORS AND TRADITIONAL DRIVING INSTRUCTION

Dr.phil. Rainer Dieterich
Professor of Psychology, University of the Federal Armed Forces, Hamburg, Germany

Abstract

This study compared truck driving instruction given to two groups who were taught to drive using driving simulators with a third group which was taught by a traditional truck driving method of instruction. The simulators were constructed by different companies and were slightly different from each other. The learning behaviors and success records of the simulator groups were analyzed and compared with the traditional group. There were eight subjects in each of the three groups (N = 24).

The major difference between the simulator groups and the traditional driver training group was that the latter group learned in actual traffic conditions. It was theorized by the researcher that the success of the experimental groups was dependent on the similarity of conditions in the simulators and in the actual truck.

Each subject in the three groups had their driving skills evaluated in a traditional driving test behind the wheel of an actual truck in traffic. Additional data on the driving skills of all subjects were obtained from rating scales and observations. The data showed that the two groups trained in the simulators often did better than the group trained in actual trucks. A discussion of the findings and an interpretation of the results will be included.

Biography

Born:	February 24, 1943 Köthen, Germany	
Education:	1969	Diplom in Psychology, University of Erlangen
	1973	Dr.phil. University of Erlangen
	1978	Dr.phil.habil. University of Munich
Employment:	1970 - 78	Assistent and Lecturer University of Munich
	1978	present Professor of Psychology University of the Federal Armed Forces, Hamburg
Publications:		8 texts, about 50 articles
Major fields:		Learning, Personality theory,

A COMPARISON OF TRUCK DRIVING INSTRUCTION USING SIMULATORS AND TRADITIONAL DRIVING INSTRUCTION

Rainer Dieterich

1. INTRODUCTION

The acquisiton of a truck driver license in Germany requires costly driver training at professional driving schools About sixty percent of truck driver licenses in Germany are given by military driving schools; this driving course lasts about six weeks. The most important part of this training involves a period of forty-five hours practicing on a military truck especially designed for driver training. This truck has two sets of controls and can be driven by either the instructor or the student. This training procedure will be referred to in this presentation as the traditional driving method.

The German military is interested in changing major parts of this training procedure by using simulators for the following reasons: (1) Ecological reasons which involves such things as avoidance of air pollution, decreasing noise, and reducing the use of fuel and other materials. (2) Reduction of traffic which is often very dense in urban areas. (3) Serious traffic jams frequently interfere and prevent driver training lessons. (4) Authorities in rural communities often prohibit driver training during rush hours.

This research is a report on an empirical study which compared traditional truck driving instruction to an innovative driver training program using simulators. The use of a simulator would eliminate problems listed above.

2. THEORETICAL BACKGROUND

The acquisition of psychomotor skills are often acquired in a procedure known as "learning by doing". For example, people learn skiing, dancing and car driving by simply doing it. Reality itself is often the most effective training medium. Any difference between the situation of learning and application, as well as differences between the learning behavior and the behavior required in real situations reduces the amount of learning. According to the theory of learning transfer, the similarity of learning and the real application determine the effectiveness of the learning.

A major concern of the study was the similarity between the truck and the simulators. Two dimensions of similarity had to be regarded: (1) The "situational similarity" of the real and the simulated learning environment expressed in terms of the technical equivalence of the simulators and the trucks and (2) the "behavioral similarity" of the learning and teaching activities of learners and instructors, expressed in terms of psychological rating scales, observational data and results of a final driving test.

3. METHOD

3.1 Subjects

24 regular students enrolled in military driving schools were used as subjects. There were eight in both of the simulator groups and eight in the traditionally trained group.

3.2 Criteria of Comparison

The following parameters of similarity were used as criteria for comparison:

Two aspects of the situational dimension were distinguished: (1) Static similarity of shape, size and the geometric relations within the learning environments and (2) the functional similarity of trucks and simulators.

Since original truck cabins were used in the simulator, the parameters of static similarity were simulated true to reality. The size of the steering wheel, the location of the dashboard instruments, seat positions were simulated

perfectly. But the technical aspects of the functional similarity required special checks. Such things as the energy required to turn the steering wheel, braking distances, the acceleration, the motornoise of the simulators were compared with those of the real trucks.

Within the dimension of behavioral similarity different kinds of parameters were identified.

(1) Psychophysical reactions of the trainees were checked by means of technical measurements; the reaction-speed before and after the learning units, stress related pulse frequency changes and weariness caused by the strain of learning.

(2) Mental processes and inner-psychic experiences of the trainees were checked. This included: learning motivation, emotional states of the trainees, risk taking behavior while driving and cognitive evaluations of the learning progress. Psychological rating scales were given to both the learners and the instructors to learn about inner-psychic reactions.

(3) Observational data recording visible reactions of the learners and their verbal interactions with the instructors were noted by observers, who sat in the cabins between the learners and the instructors. The verbal teaching behavior of the instructors was viewed as an indicator of learning. A special code was developed to record the verbal instructions related to the learning behavior. Since the ability to distinguish obervations is limited, the criteria had to be carefully defined, and the observers thoroughly trained. A distinction was made between formal and content aspects of teaching. Three types of formal aspects were combined with three types of content aspects, thus gaining nine combinations of encoding categories plus some additional categories of driving mistakes.

Definition of terms used in the oberservations: Formal categories: (1) Hints: verbal comments directing the learners' attention before a driving action takes place, (2) Corrections: instructor comments following failures of the learners, (3) Reinforcements: comments on positive or skillful driving actions intending the repetition of the same behavior in future situations.
Content categories: (1) Operation of instruments: comments on the use of such things as handling of gas, gear, brakes, indicators, (2) Perception: comments on traffic and environment observation outside the cabin, (3) Way of driving: comments on the speed, carefulness, distances etc.

In case of corrections the observers also encoded the kind of mistake the correction was related to.

3.3 Experiental design, learning objectives and contents

The learning objectives were taken from the official curriculum of German military driving schools.

A design was developed for this experiment which included nine thirty-minute learning units; that is, each subject spent this amount of time in either the simulator or the truck. Three of the units were devoted to general driving without pursuing special objectives, and six units were devoted to the additional pursuit of special objectives. The units, completed by a final driving test were arranged according to the following sequence:

Unit 1: General driving
Unit 2: General driving plus starting on hills, stopping at high speeds, use of lanes.
Unit 3: General driving plus changing lanes, passing, keeping proper distance.
Unit 4: General driving
Unit 5: General driving plus selecting proper lanes, turning right or left, correct behavior at crossings.
Unit 6: General driving plus uphill and downhill driving.
Unit 7: General driving
Unit 8: General driving plus driving through narrow lanes, obeying rights of way on turns
Unit 9: Approaching ramps, backing a trailer

Each of the subjects had to complete three learning units every day. Together with the initial

pretest and the final posttests the entire procedure took five days to complete.

3.4 Statistical Data Analysis

The rating values and observational data were compared and analyzed by means of a two factor ANOVA (Analysis of Variance). The ratings and observational frequencies were used as dependent variable, the groups (simulator- and traditionally trained group) as the first factor and the learning units as the second factor.

This method revealed the statistical significance of Differences between the compared groups, of learning effects within the course and of Interactions between groups and learning units, that is differences in driving skills during the course, subjected to simulator or traditional training.

The final driving test results for all subjects (18 subscores) were compared using t-tests.

4. RESULTS: EXAMPLES AND INTERPRETATION

4.1 Rating Scales taken from the Instructors

Scale "tired - alert": This example demonstrated how similar the simulator group and the traditional group can be (see figure 1).

There was no significant difference between the groups ($p = 0.415$) but significant differences were found among the learning units ($p = 0.001$). The comparison shows that there was a decrease of alertness as a consequence of the physical and mental strain from the first to the third learning unit every day. Furthermore it shows that there was a reduction of that decrease from the first day to the third day as a consequence of training and increasing familiarity. These extremely similar results indicate that physical and mental strain were the same within the simulator and the real truck.

Scale "risk taking - careful": According to the criterion of risk taking or careful driving a difference seems to exist (see figure 2).

There is no significant difference within the driving units ($p = 0.209$) but a highly significant difference between the groups ($p = 0.000$). It is obvious that the control group was driving more carefully. However, this finding is more complex than it appears to be; the final driving test indicates that the simulator group drove more carefully than the traditional group.

A possible explanation is that the values of the simulator group were very constant. They were keeping in the neighborhood of the neutral point of four. In that case the neutral value is the ideal one. That means, the learners were neither risk taking nor over- cautious. It may be that this behavior was more "realistic" than the over-cautious behavior of the traditional group. Perhaps trainees learn better if there is no reason to be over-cautious. This way the learners have a chance to learn by mistakes, to correct mistakes instead of avoiding them and to finding out what they can and cannot do.

A further finding is that there is no indication that a simulator causes a learner to take more risks. In fact the findings suggest that both groups became more careful in the course of training.

4.2 Self rating scales taken from the learners

Scale "calm - nervous": The comparison shows that the simulator group was less nervous than the traditional group except in the first learning unit (see figure 3).

There was no significant difference among the learning units ($p = 0.145$) but a highly significant difference between the groups ($p = 0.000$). Even though this result shows that the simulator does not create as much stress as an actual truck, this finding indicates another advantage of the simulator. A person often learns better if the situation is relaxed and free from stress. This might also account for the better scores the simulator groups obtained in the fin[illegible] driving test.

Scale "anticipation: look foreward - don't look foreward": This comparison showed that the learning motivation as expressed by the enjoyment of driving a real truck was much higner than the pleasure of learning with a simulator (see figure 4).
There was no significant difference among the learning units (p = 0.926) but a highly significant difference between the groups (p = 0.000). Truck driving seems to be one of a soldiers' most enjoying, motivating and stimulating activities. There is no chance to motivate him by a simulator in a similar way. Those theories which say that simulator learning is a very motivating kind of learning may be accurate in many instances, but truck simulators cannot compete with a real truck. But on the other hand the graphic shows that the learning motivation within the simulator is high enough to produce efficient learning.

Scale "efficacy of learning: low -high": The comparison shows that the simulator learners were rather skeptical at the beginning of their training. They thought that they were learning less than the traditional group (see figure 5).

There was a slight but non-significant increase of the confidence of the simulator group (p = 0.367) showing, that the subjects became more and more convinced that they were learning effectively. The curve indicates a continious reduction of scepticism, but this result may not be generalized because of the lack of significance. It is interesting to note that the simulator group was actually learning more efficiently than the traditional group.

4.3 Observation scales

Scale "total amount of hints": Within both groups verbal teaching activities seem to be concentrated in the first part of the course. The decrease of those curves was even larger in the simulator group than in the control group (see figure 6).

There is a highly significant effect among the learning units (p = 0.000) but no significant difference between the groups (p = 0.701), which means that the continuous decrease during the course actually existed in both groups, but that the stronger decrease within the simulator group may only have existed in that sample and may not be generalized to the population. However the results indicate that more teaching aids were required at the beginning of the training period in the simulator than in the real truck. After the third learning unit there was less instructor involvement in the simulator group. After having understood the introducing information the learners were able to learn more independently. They were able to gain experience by themselves, to learn by mistakes and to have learning experiences not possible in a real truck.

Scale "verbal activities of instructos": Altogether the simulator group obtained more reinforcements than the traditional group (see figure 7).

Both kinds of effects were significant; differences were found among the learning units (p = 0.028) as well as between groups (p = 0.015). This may also be attributed to the concentration of the teaching activity in the beginning of the training period. The higher number of reinforcements indicates an advantage of the simulator because reinforcement is usually associated with positive learning.

4.4 Mistakes of the Learners

The criterion of mirror observation is an example of a simulator characteristic which needs theoretical interpretation (see figure 8).

It is obvious that the simulator group had more problems with the correct observation of the mirrors. The differences between the groups were highly significant (p = 0.000) as well as those among the learning units (p = 0.000). An explanation may be that the mirrors in the simulator did not give the learners the information they wanted and expected. There was so little traffic that the learners did not expect other traffic participants anyway. If the learners did not see what they expected, they did what every driver does: They were looking behind them through the window, they were

turning or moving their their bodies in order to look at the mirror from a different angle, but this act was useless and only confused them. This may be why they made more mistakes during the driving lessons. At first glance this appears to be a disadvantage of the simulator, but the final driving test showed that the simulator groups were better able to observe the mirrors than the traditional group. The reason for this is quite simple.

Even though it is more difficult to acquire that skill of mirror observation in simulators, the trainers insisted on it. This is a training effect caused by additional complications. Like a sportsmen who wears a leadbelt during the training process and who jumps higher or runs faster when he gets rid of it in the real competition, the simulator learners overcompensated for this handicap. These results indicate that the entire system of learner, instructor and medium function in a holistic way. An evaluation has to involve all three facets. A disadvantage can be turned into an advantage by means of didactical learning aids.

4.5 Final Driving Tests

The final driving tests were given by regular test officers. They used rating scales as criteria of evaluation (see figure 9).

According to the t-test-results, (Separate-Variance-Estimates of 2-Tail-Probabilities), there were significant differences between the groups, which indicate higher scores in the simulator groups.

There were advantages of the simulator learners according to the criteria of mirror observation ($p = 0.049$), keeping correct distances ($p = 0.007$), correct use of lanes ($p = 0.015$), stops and starts ($p = 0.043$) and turning right/left priorities ($p = 0.011$). These factors appear to be associated with careful and safe driving.

5. SUMMARIZING INTERPRETATION

In summary, truck driving taught by simulators was more effective than the traditional method. Even though the simulators were imperfect prototypes and in the process of further development and technical improvement, their success was obvious. However they still had some disadvantages, including the simulation of the mirrors, the motor noise or the occurrence of motion sickness. One of the major problems was the simulation of heavy urban traffic. But the main result of the study was that these disadvantages did not seriously affect the learning. Obviously it is not necessary to have perfect full task simulators in order to obtain efficient learning processes. The reasons for the better results of the simulator group compared to the traditional group may be that:

- Simulators provide a very intensive kind of learning. Specific situations or problems can be reproduced artificially as often as on likes. The learner does not depend on a real traffic situation.

- There are training effects due to additional complications like that "leadbelt-effect", mentioned above. That is, simulator handicaps can be overcompensated by means of enhanced didactical support from the instructor.

- Simulators allow the separation of different aspects of the learning process from each other instead of mixing and contaminating them like the real situation. In the real traffic the learner has to observe the mirror, the instruments and the traffic simultaniously. At the beginning of the learning process any of those activities may disturb each other. Simulators permit the learner to perform such activities one after the other, and then to finally combine them and thus to avoid their mutual contamination.

Figure 1: Comparison: Truck-Simulator
Trainer - Rating

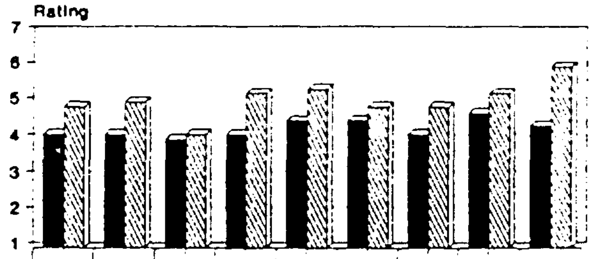
	1	2	3	4	5	6	7	8	9
Simulator	5,5	4,75	4,143	5,5	4,875	4,5	5,25	5	4,875
Truck	5,125	4,75	3,5	5,125	5	4,5	5,25	5	4,875

Learning Unit

Simulator Truck

tired - alert

Figure 2. Risk taking - careful
Trainer - Rating



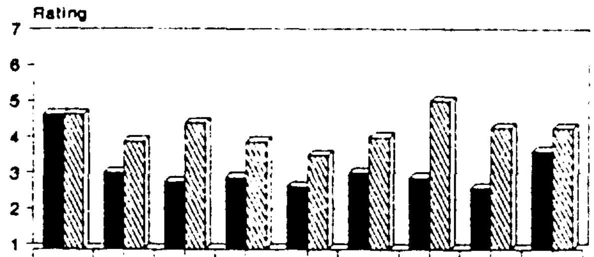
Learning Unit	1	2	3	4	5	6	7	8	9
Simulator	4,125	4,125	4	4,125	4,5	4,5	4,125	4,714	4,375
Truck	4,875	5	4,125	5,25	5,375	4,875	4,876	5,25	6

Learning Unit

Simulator Truck

risk taking - careful

Figure 3: Scale: calm - nervous
Learner Self Rating



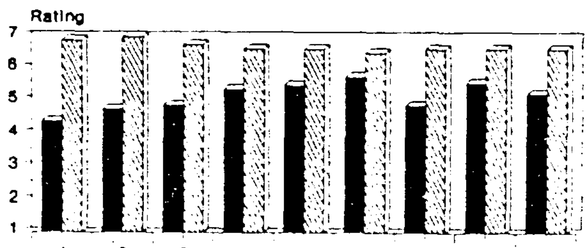
Learning Unit	1	2	3	4	5	6	7	8	9
Simulator	4,75	3,125	2,875	3	2,75	3,126	3	2,714	3,75
Truck	4,75	4	4,5	4	3,625	4,125	5,126	4,375	4,375

Learning Unit

Simulator Truck

calm - nervous

Figure 4: Scale: Anticipation
Learner Self Rating



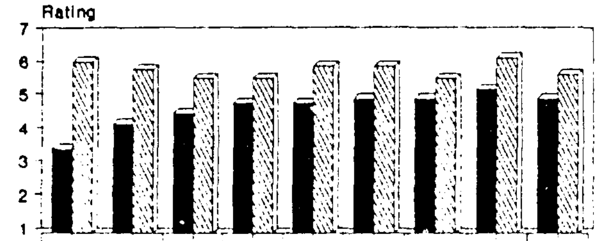
Learning Unit	1	2	3	4	5	6	7	8	9
Simulator	4,375	4,75	4,875	5,375	5,5	5,75	4,875	5,571	5,25
Truck	6,875	6,975	6,75	6,625	6,625	6,5	6,625	6,625	6,625

Learning Unit

Simulator Truck

Anticipation:
look foreward - don't look foreward

Figure 5: Scale: Efficacy of Learning
Learner Self Rating



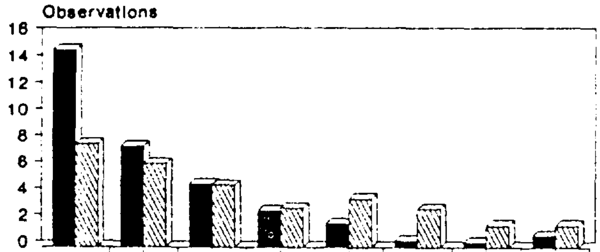
	1	2	3	4	5	6	7	8	9
Simulator	3,5	4,25	4,571	4,875	4,875	5	5	5,286	5
Truck	6,125	5,887	5,625	5,625	6	6	5,625	6,25	5,75

Learning Unit

Simulator Truck

Efficacy of Learning
low - high

Figure 6: Total Amount of Hints
Verbal Activities of Instructors



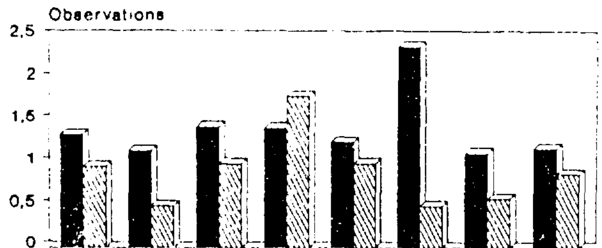
	1	2	3	4	5	6	7	8
Simulator	14,793	7,571	4,714	2,708	1,75	0,5	0,375	0,375
Truck	7,751	6,25	4,625	2,916	3,625	2,875	1,624	1,625

Learning Unit

Simulator Truck

total amount of hints

Figure 7: Comparison Simulator - Truck
Verbal Activities of Instructors



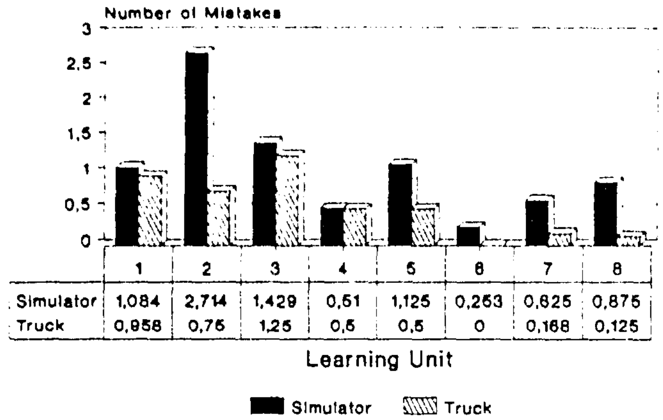
	1	2	3	4	5	6	7	8
Simulator	1,333	1,143	1,429	1,416	1,25	2,375	1,125	1,174
Truck	0,98	0,5	1	1,793	1	0,5	0,584	0,875

Learning Unit

Simulator Truck

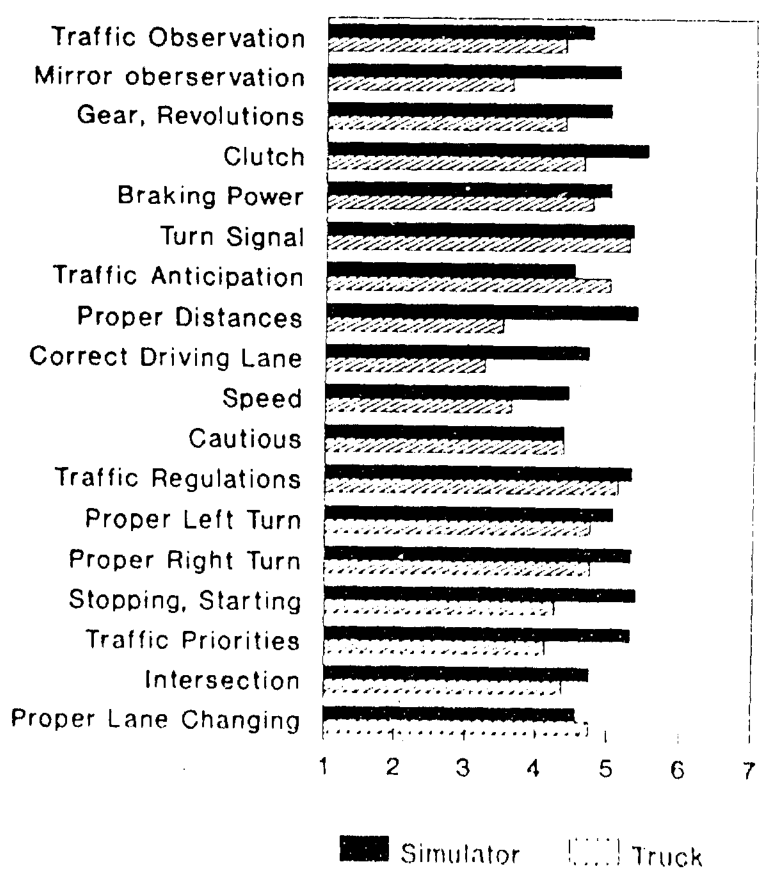
Reinforcements of driving manner

Figure [illegible]: Comparison Simulator - Truck
Mistakes of Learners



	1	2	3	4	5	6	7	8
Simulator	1,084	2,714	1,429	0,51	1,125	0,253	0,625	0,875
Truck	0,958	0,75	1,25	0,5	0,5	0	0,168	0,125

Figure 9: Comparision Truck - Simulator
Final Driving Test



7 very good, 1 very bad

THE PROs AND CONs OF THE USE OF NDI SOFTWARE ON GOVERNMENT CONTRACTS

Nathaniel R. League
Michael F. Wilt
AAI Corporation
Hunt Valley, Maryland

ABSTRACT

The application of NDI is becoming more wide spread, especially with the development of better software development methodologies, such as Object Oriented techniques, and languages, such as Ada. Contractor's who have a substantial quantity of NDI software for a specific application, for example, flight simulation, radar simulation, etc., maintain a significant advantage over those who do not when responding to a Request For Proposal from the Government. Their cost can be substantially lower than others and thus provide the contractor with the most available NDI software for the required application, an "easy" win.

Unfortunately, the Government's definition of NDI continues to baffle many as to exactly what qualifies as NDI. This typically leads to long and generally somewhat unresolved battles between program managers and engineers on both sides following the award of the contract. Typically the contractor may have a difficult time qualifying his perceived NDI software as NDI software. Also, the Government still requires a substantial degree of design, performance, implementation, and testing information relating to the NDI.

ABOUT THE AUTHORS

Mr. League is the Manager of Systems Engineering with the Training and Test Division of the AAI Corporation. Mr. League has been assigned as the systems and project engineers on many of AAI's simulation programs. Mr. League holds a B.S. degree in Computer Science from The University of Maryland. Mr. League has also completed course work for an M.S. in Engineering Sciences from Loyola College. Mr. League has 23 years of experience working with real-time software in the fields of signal processing and simulation systems.

Mr. Wilt graduated from The Johns Hopkins University in 1965 with a B.S. degree in Mathematics. He has been employed at AAI Corporation since 1961. Mr. Wilt's primary role at AAI has been in the field of computer software for numerous types of simulation systems, including Electronic Warfare Trainers for the US Air Force, Anti-Submarine Warfare Trainers for the US Navy, Unmanned Systems Controllers, and Digital Radar Landmass Simulators. Mr. Wilt has been involved in all aspects of simulation software development, including management, design, code, test, integration, and documentation. Mr. Wilt is currently the Project Engineer on various DRLMS programs at AAI.

THE PROs AND CONs OF THE USE OF NDI SOFTWARE ON GOVERNMENT CONTRACTS

Nathaniel R. League
Michael F. Wilt
AAI Corporation
Hunt Valley, Maryland

INTRODUCTION

The use of nondevelopmental item (NDI) software on Government contracts may seem straightforward. The Government can save money, schedule and obtain a more reliable product in the process. The contractor can be more competitive by having a product on the shelf that their competitors do not possess. There are many problems, however, which present themselves when NDI software is used on a Government contract.

o Often the Government and the contractor do not agree as to what qualifies as NDI software.

o There are often differences of opinion as to the level of documentation that should be provided with NDI software.

o There are disagreements as to what rights the contractor keeps and what rights the Government inherits with the NDI software.

These issues have led to many heated discussions on Government contracts. One of the main causes of these differences of opinion is that the definition of NDI software is too vague, thus, the contractor often assumes that almost any piece of software developed using their money can qualify as NDI. Government representatives, however, do not always agree with the contractor's declaration that a certain piece of software is NDI.

This paper does not intend to conclude whether NDI software should be encouraged within the Government procurement process. The Government has already decided that NDI software is to be an integral part of its procurement philosophy. The authors are completely in favor of using NDI software in Government procurement; however, there are many problems that occur with the use of NDI software. Before the advantages and the problems of using NDI software are discussed, it is necessary to examine the existing definition of NDI software.

DEFINITIONS OF NDI SOFTWARE

Government definition. There are three definitions for NDI software found in the following Government documents:

DOD-STD-2167A
MIL-STD-973
MIL-STD-480B

DOD-STD-2167A. In this Department of Defense (DoD) standard, the Government defines nondevelopmental software (NDS), which is the same as NDI software, as follows:

"Deliverable software that is not developed under the contract but is provided by the contractor, the Government, or a third party. NDS may be referred to as reusable software, Government furnished software, or commercially available software, depending on its source."

MIL-STD-973. In this standard, the Government defines nondevelopmental items as follows

"Non-developmental item is a broad generic term that covers material available from a wide variety of sources with little or no development effort required by the Government. NDIs include:

a. Items obtained from a domestic or foreign commercial marketplace.

b. Items already developed and in use by the Services, other defense activities, and Government agencies.

c. Items already developed by foreign governments which can be supplied in accordance with mutual defense cooperation agreements and Federal and DoD acquisition regulations. (SD-2)"

MIL-STD-480B. This standard defines nondevelopmental items as follows:

"Non-developmental items are existing developed and available hardware or software that are capable of fulfilling DoD requirements, thereby minimizing or eliminating the need for costly, Government-sponsored research and development (R&D)programs. An NDI is usually an off-the-shelf or commercial type product, but may also include hardware or software already developed by or for the DoD, or other military services or foreign military forces."

Certain key phrases imply that any software developed by the contractor can qualify as NDI. These phrases have been underlined. All three of these standards indicate that any software developed by the contractor without Government funding is NDI. MIL-STD-973 also indicates that some alteration to the NDI software package is permissible. This alteration of the NDI software using Government funds has been the basic cause of many of the differences in opinion between Government and contractor personnel. How much alteration is "some alteration"? Some people feel that as much as 50 percent alteration of NDI is allowable before the software loses its NDI qualification, others feel that no alteration is allowed.

If NDI software is so difficult to define and causes so many problems, why do both Government and contractor personnel insist on using it on Government contracts?

REASONS FOR USING NDI SOFTWARE

There are many advantages for using NDI software. Some of these advantages are primarily from the Government perspective, others are more beneficial to the contractor, while still others benefit both the Government and the contractor. These advantages are offered below for consideration.

Reduced cost. In our current environment of reduced operating budgets, this is probably the single biggest advantage for using NDI software. From the Government's point of view they simply save money on their procurement. NDI software allows the contractor to allocate his development costs over several contracts, commercial as well as Government; therefore, no one contract has to bear the entire cost of the NDI software development. The big advantage to this cost savings from the contractor's point of view is that it gives them a competitive advantage over other bidders. Software development costs are high. In today's world, software costs, particularly on one-of-a-kind systems, are often much higher than any other cost on the project. If the contractor has unique NDI software sitting on the shelf which can be used to solve the customer's problem, the contractor can sell that product to the customer at a price that is substantially less than the development cost of the product.

Reduced schedule. Software is often the real schedule driver on a new development project. If a substantial amount of the software for the contract can be obtained as NDI, the product

development time can be significantly decreased. The advantage of this reduced schedule from the Government's perspective is obvious. The Government often needs to have systems fielded much sooner than is possible because of the development time required for new, one-of-a-kind products. If NDI software allows a contractor to field a system sooner than otherwise possible, the Government would generally be quite pleased. Because the majority of contractors are in business for the purpose of making a profit, the advantage to them is that their costs are further reduced as the schedule is reduced. There are certain overhead costs that are linearly tied to schedule duration; thus, if the contractor can reduce schedule, he can reduce cost.

High reliability. The Government has the right to have certain expectations concerning NDI software. They have the right to assume that the software has been properly developed, has been properly tested, and is ready to be delivered or modified (if modification is required). If the NDI software meets these expectations, the Government has the right to expect that the software works properly and has been even more thoroughly tested than what they would expect on a newly developed software package. One would expect to find few or no problems with the NDI software package during testing or after it is fielded.

Reduced life cycle costs. The costs of maintaining and supporting NDI software should be much less than those for newly developed, unique software. The product should be mature, which will reduce the cost of finding and correcting errors. The product should also be isolated enough from the unique software such that it does not need to change when the unique software changes.

Encourages companies to invest. Another advantage of NDI software is that it encourages contractors to invest their own money in developing "off-the-shelf" software packages. As mentioned earlier, the contractor does this to gain the competitive advantage that he seeks.

Improved efficiency. NDI software is developed to the contractor's standards. The software can be made more efficient with respect to computer resources, (i.e., CPU time, memory, and disk space). The programming language that best fits the need can also be used (e.g., "C", Ada or assembly language).

These are the advantages of using NDI software. Obviously, there must be some problems associated with the subject or there would be no need for this paper. Some of the problems experienced by AAI are presented below.

PROBLEMS IN USING NDI SOFTWARE

Though there are many advantages to using NDI software, there are also some problems associated with its use. These problems vary in severity depending on the observer's perspective. Some of the problems presented below are purely from the Government's point of view while others are from the contractor's point of view.

Qualification of NDI software. This is typically a major problem from the contractor's point of view. Imagine that you were just awarded a contract where you assumed that 80 percent of the software on the contract would be classified as NDI software. Further assume that as the contract progresses, you will need to modify the NDI software by approximately 20 percent.

You now have the makings of a real NDI problem. Your customer is most likely not going to want to qualify your software as NDI because you are using Government money to significantly change your company-owned NDI software. There is a

good chance that five things are going to happen to you now.

1. Your customer is going to want to review the entire software package which means that you must have design reviews with the customer for this so called NDI software.

2. Your customer is going to want a complete set of software documentation for the software package to whatever standard is required by the contract.

3. Your customer will test the complete software package just as if it were a new development.

4. Your customer is going to want the rights to the software because he has paid for its modification. This means that the customer can give the software to other contractors if it is required for them to modify your software at some future date.

5. You are probably going to lose money on at least this portion of the contract.

Documentation may be poor. One problem faced by the Government with NDI software is that the software is developed and documented to the contractor's standard. Depending on the contractor, this standard can be acceptable or unacceptable. The contractor, however, certainly expects the Government to accept the documentation in whatever format and to whatever degree they have provided.

Development standards. Government personnel often feel uneasy in accepting NDI software that was developed to a standard that is invisible to them. The contractor's position is that if the software works, it does not matter how it was developed. The Government has repeatedly demonstrated that they are uncomfortable with this attitude.

Testing. The contractor generally feels that if a software package is NDI software, it should be exempt from some of the rigorous testing that would be performed if the software were uniquely developed for a particular contract. After all, how many people do a thorough test of a compiler or a word processor package before they accept (buy) it? The Government's attitude is usually that the contractor is not MICROSOFT, that the software is not as mature as the items listed in the examples, and that they, the Government, want to test the software before they accept it. Again, this has led to more than one interesting discussion.

Data rights. Remember our poor program manager with the NDI software package. After, the Government has decided that the modified NDI is no longer NDI, they want complete rights to the software. After all, they are funding a significant portion of its development.

What we need in the industry is a more rigorous definition of what is required for a software package to be considered NDI.

NEED FOR A BETTER DEFINITION/UNDERSTANDING OF NDI

All of the advantages discussed above are reasons enough to justify the use of NDI software on Government contracts; however, the problems show that there are some issues that need to be resolved.

We need a better definition of what constitutes NDI software. In the minds of some people, NDI software means that the offeror has a completed software package that has a clear interface to other hardware and software components and that will not require any modification during the life of the subject contract. To others, at the opposite extreme, NDI software is any software package that they (i.e., the contractor) developed to be

interfaced with other non-NDI in any way required. What most people consider as NDI software falls in between these two extremes.

Based on the Government's own definitions given at the beginning of this paper, either of the software packages in the above example can qualify as NDI. After winning a contract, however, the program manager may find that it is quite difficult to get a software package qualified as NDI.

The rules for NDI software need to be explicitly defined. Does the software have to be unmodified to be considered as NDI? If it can be modified, to what extent can it be modified before it loses its NDI status? If the NDI software is modified, does it lose any of its NDI advantages? This issue is not for us to decide and is not really the subject of this paper, however, the following must be considered. There is a balance between allowing only mature, stand-alone software packages to qualify as NDI and allowing any previously developed contractor software to qualify as NDI. As the requirements for NDI qualification become tougher, the Government may receive less NDI on its contracts and will, therefore, eliminate some opportunities for cost savings. If, however, the Government accepts virtually anything as NDI, they will receive a lot of software that is not properly developed, documented and tested. This will result in the Government receiving substandard products.

It is extremely important from the contractor's point of view that the proposal team know the rules for NDI software. It is not fair for a proposal team to bid one price for a development effort based on their assumption as to what will be NDI and what will be developed, then find out after contract award that their assumptions are not acceptable to the Government project team.

SPECIFIC EXPERIENCE ON THE DRLMs PROGRAM

AAI has had some very recent experience with the trials and tribulations of NDI software. AAI has been under contract with the USAF to build a software intensive Digital Radar Landmass Simulator (DRLMS). It was known by both the government and AAI at the onset of this contract, that most of the software which AAI would be using on this program had been developed by AAI using IR&D funds. This software has become known as the "core" DRLMS software because it performs the basic radar functions required by any software intensive DRLMS.

Early in the program, AAI and the government established that a significant portion of the "core" DRLMS software would be used on the program and that this software would be categorized as NDI. Other software developed on the program would be trainer unique and the Government would retain ownership of this software and its documentation. Also, these other, non-NDI, items would be documented in accordance with the contract requirements, which meant a Software Design Document (SDD), and a Software Product Specification (SPS) would be developed by AAI.

In order to establish some sense of continuity in the SDD, AAI agreed to provide a high level description of the NDI software in the SDD, while providing a fully specification compliant SDD for the unique software. In the SPS, listings would be provided only for the trainer unique software.

As the program progressed, AAI found the need to make modifications to the established NDI software. The question was immediately asked "Is this code still NDI?" There was no simple answer to the question. There were no established guidelines as to what degree of modification transforms NDI software into trainer

unique software. Herein lies one of the basic problems with using NDI software.

AAI maintained that, while in some cases the software was being significantly changed, the basic algorithms remained unchanged; and therefore, the modified software still qualified as NDI. This disagreement persisted throughout most of the program, even up until the physical configuration audit (PCA). Fortunately, for AAI, an agreement was reached with the Government and the issue was eventually settled.

Had the Government representatives been less reasonable or less willing to work with the AAI team , AAI could have incurred a significant schedule delay with an associated cost overrun.

PROBLEMS AND RECOMMENDATIONS

In spite of the problems encountered using NDI software, the concept is certain to stay. The issue of what is NDI software and what does it mean to be NDI software may not be resolved for a long time. There are some lessons which have been learned by the Government and industry. Some of the problems encountered and a recommendation of how to prevent them is presented below.

PROBLEM. Using nondevelopmental item software may seem like a panacea. But, there are many pitfalls that can make the experience extremely unpleasant.

RECOMMENDATION. The Government and contractor personnel need to work together early in the procurement cycle to reach an understanding concerning NDI software. The contractor needs to make their intention known, and the Government needs to let the contractor know its position on the contractor's intent.

PROBLEM. Often the Government and the contractor do not agree as to what qualifies as NDI software.

RECOMMENDATION. If the contractor intends to use NDI software on the contract, let the Government project team know as soon as possible (i.e., during the proposal phase). Come to an agreement as to the acceptability of the software as NDI before preparing the final cost estimate.

PROBLEM. The alteration of NDI software using Government funds has been the root of many differences in opinion between Government and contractor personnel as to whether the software package is still NDI.

RECOMMENDATION. This is probably the most serious problem you will face. Be sure to let the Government know up front if you intend to modify your NDI software to meet the contractual requirements. Reach an agreement that both parties can live with before contract award.

PROBLEM. If a software package qualifies as NDI, the Government may still expect certain considerations pertinent to the software package. They may require documentation and design reviews of the software package. The Government may also want to acquire certain rights to your software package even though it qualifies as NDI.

RECOMMENDATIONS. Communicate with your customer. Make sure that you and the customer are in agreement as to what design reviews , if any, will be held, what documentation will be provided and what rights are being given to the Government. Make sure you reach an agreement before contract award.

PROBLEM. The existing definitions of NDI software leave too much latitude. Government and contractor project personnel can both be

talking about NDI software, thinking that they are talking about the same thing when, in fact, they are not.

RECOMMENDATION. We need a better definition of what constitutes NDI software. The definition needs to address NDI software qualification, documentation, and testing standards. The definition needs to address modification of NDI software using Government contract funds and to define who owns the NDI software and what it means to own the software.

RECONFIGURABLE TRAINERS IN SOFTWARE LIFE CYCLE MAINTENANCE

Neil F. Cosgrove
INFOTEC Development
Sparta, New Jersey

ABSTRACT

One way of achieving these seemingly contradictory goals of military preparedness within the new economic reality is through computer simulation training. Despite the fact that simulation can result in great cost savings, even new simulation efforts will face increased fiscal pressure. One means to reducing software cost is to maximize the use of development resources. Traditionally, software development and maintenance have required the use of unique hardware development platforms. The availability of these platforms is often the gating factor in development. Yet the use of these facilities is often cyclic, periods of frantic activity where all resources are fully committed to periods of no activity where resources are sitting inactive.

In this paper, we will discuss our experience of this problem as pertains to software maintenance of the U.S. Army's Conduct of Fire Trainers (COFTs) for the M1, M1A1, and Bradley fighting vehicles. Traditionally, software development for these systems had implied the dedication of one or more of the actual target training systems. This has posed the problem that the required COFT system was not always available during software development. Our solution to this challenge was to develop a reconfigurable COFT trainer for software development. Realizing that real world fidelity is a training concern not required for software development, we have developed a family of devices that allow any COFT of the above cited vehicle types to simulate any one of the other vehicle types. This paper provides our lessons learned from our experience that we believe will have broad range applicability not just in software maintenance, but in the development of future training systems.

Neil F. Cosgrove: Received a Bachelors Degree from Manhattan College, Riverdale N.Y. in 1980 with majors in Physics and Mathematics. Neil received his Masters of Computer Science Degree from Polytechnic Institute of New York in 1983. Neil has over ten years of software development experience for defense related applications. Prior to joining Infotec Development, Neil was a Member of the Technical Staff at the MITRE Corporation for five years where he worked on the systems architecture for the Integrated Attack Warning and Assessment Architecture system at Cheyenne Mountain Air Force Base. Upon joining Infotec Development, Neil has been the Software Team Leader for the M1 Abrams Conduct of Fire Trainer (COFT) software block upgrade.

RECONFIGURABLE TRAINERS IN SOFTWARE LIFE CYCLE MAINTENANCE

Neil F. Cosgrove
INFOTEC Development
Sparta, New Jersey

INTRODUCTION:

While the lessening of military threats in the "New World Order" is debatable, the austerity imposed on defense budgets in the age of "Economic Reality" is not. The government will be adverse to the funding of new initiatives, yet will continue to demand the same level of preparedness from its military forces and the ability to master any threat with minimum loses.

One way of achieving these seemingly contradictory goals is through computer simulation training. Yet, despite the fact that simulation can result in great cost savings, it would be naive not to believe that even new simulation efforts will face increased fiscal resistance. As the actual weapon systems are asked to "soldier on" for longer periods of time, so to shall the corresponding training systems. This shall place new emphasis on the maintenance and upgrade of the most expensive part of the training system: the software.

One means to reducing software cost is to maximize the use of development resources, and that is the subject of this paper. Traditionally, software development and maintenance have been based on the use of unique hardware development platforms. Frequently a target system(s) has been dedicated to this purpose. The availability of these platforms is often a gating factor in development. Yet the use of these platforms is often cyclic, periods of frantic activity where all resources are fully committed to periods of no activity where resources are sitting idle. This puts the software development manager in a dilemma: software development productivity could be increased if more development platforms were available, yet the manager realizes that the period of effectiveness in the development process for these platforms is short thereby making the justification for having additional development platforms difficult.

An analogous situation has been faced by anyone who likes to work on their house or car. Eventually, one is faced with a job that can not be done for lack of the proper tool. Often a specialized tool for the task exists, but the do-it-yourselfer knows that once the current job is completed, this specialized tool, often bought at a premium, is likely to sit in the tool box for an extended period of time before another application for it is found. The more practical approach is usually to buy an attachment to reconfigure an existing tool. This allows us to complete the job at hand, yet the basic tool's functionality is preserved by the removal of the attachment. The attachment can often be obtained at significant cost savings over the specialized tool. The cost of the attachment is usually more than paid for by the ability to accomplish the task and having the attachment in ones tool box now becomes a bonus by increasing the productivity of the original tool.

Why can we not map this common sense concept to software maintenance? By reconfiguring, through non-destructive attachments, existing and under utilized hardware platform, A, to function as a different development platform, B, where B is an over subscribed development resource, we can increase the productivity of the software maintenance effort on B without sacrificing our platform A capabilities.

AN OVERVIEW OF THE PROBLEM:

Software Maintenance on the Current COFT Systems

As an example of the problem, let us examine the current issues involved with the Conduct of Fire Trainers (COFT) for the M1, M1A1 Abrams Main Battle Tank and the Bradley Fighting Vehicle. The mission of these trainers is to train precision gunnery skills for gunners and commanders either individually or as a crew. Despite the fact that the COFTs for each system is managed as a separate configuration item, all three trainers have evolved from a common origin and share similar characteristics and components: an Image Generator for training visuals, a Processing Engine, a crew

compartment which simulates a subsection of the appropriate vehicle and is driven by the synthesis of the outputs of the Image Generator and Processing, and an Instructor/Operators Station (IOS) where the Instructor/Operator (I/O) monitors and controls the simulation. The IOS, which will become particularly important in the latter sections of this paper, comprises a terminal interface to the processing engine for simulation control, student scoring, and system management functions and visual monitors which "echo" what each student is presented through his various sights. A simplified block diagram of the COFT system is presented (see Figure 1).

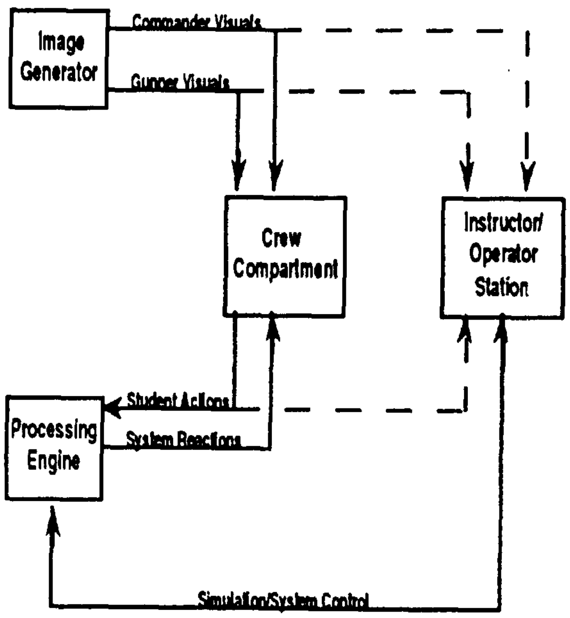


Figure 1

The COFT systems in question have been providing effective crew training for approximately ten years. As would be expected, the software has required on-going maintenance, both to address the perennial software bug and to add new features to reflect new and evolving doctrine that must be imparted to the students. These changes are encapsulated in Software Block Upgrades (SBUs) which usually reflect a collection of maintenance items that have accumulated over the course of two to four years. The average suspense for completing and fielding a new software baseline is on the order of twelve to eighteen months. Many of these changes are human factors related, in particular how information is presented in the form of visuals. In many ways this becomes more a matter of art than engineering, and often requires extensive "hands on" development time to achieve the best result from a human perspective.

Utilization Cycle of COFT Platforms in Software Development.

The problem is how does one get the system time to do this "hands on" analysis, design, and test. The traditional answer has been to have an actual COFT system at the Life Cycle Support Center. However, this is only a partial solution. Available COFT systems are a rare and precious commodity and their primary mission is training, not software maintenance. This has resulted in cases, as in the M1A1, where due to all M1A1 COFTs being fully committed to their training mission that there simply was not an M1A1 COFT available as a software testbed during the time frame when an SBU was to be accomplished.

Even if a COFT is available for Software Maintenance, one must question the cost effectiveness of dedicating a system to this purpose full time. As those familiar with the unwritten laws of software development know, software development essentially occurs in spurts: periods of frantic activity where all the developers need the system followed by lulls in which the system sits idle. Despite best efforts of program managers to minimize this phenomenon, this corollary of Murphy's Law appears to be inviolable. These problems become further exacerbated when one attempts to maintain more than one type of COFT. Now space, cooling and power must be allocated to support an M1 COFT testbed and a Bradley COFT testbed. However, experience has shown that one never has the "right COFT" when you need it. When there is a "hot" Bradley job and every developer needs access to the Bradley COFT, the Abrams COFT is often under utilized and vice versa. These facts place the program manager on the horns of a dilemma: If I had more COFTs I could gain greater productivity from my human resources and reduce cost and schedule risk to my SBU program. However, I can not reasonably expect to divert COFT systems

to Software Maintenance when there are not enough to support the training mission, nor is it cost effective to support and maintain these systems during the lulls in software development.

How Much Fidelity is Required for Software Maintenance?

To find a solution to this problem requires that we revisit the function block diagram of the COFT system as shown in Figure 1. From this viewpoint, we are reminded that the COFT systems have more in common, i.e., the Image Generator, Processing Engine, and IOS, than they have differences, i.e., the crew compartment and the actual software that resides on the compute engine. The software is an easy issue to tackle in that it requires merely a change of disk pack and a reboot, and the processing engine is then executing the new vehicle code. The tougher issue is the crew compartment. The crew compartment of an Abrams Main Battle Tank is drastically different from that of a Bradley Fighting Vehicle as they represent different weapons, fire control systems, sights, and so on. However, for engineering purposes do we need full fidelity? The answer is no, we as engineers do not need to develop the reflexes and muscle memory that an actual armored fighting vehicle crewman needs to develop. Instead, the engineer requires a means to supply a logically equivalent input to excite the software and then a means to observe the resulting output. The IOS already provides the means of viewing the system output without the space constraints of actually having to sit inside a crew compartment. All that is required is a means of applying the functionally equivalent input.

Again, when one looks at the crew compartment from an engineering perspective it becomes apparent that the input devices are functionally equivalent to simple switches and it is therefore possible to replicate the crew station on a **purely logically equivalent**[1] basis by a collection of simple switch boxes. Our system block diagram for engineering purposes now becomes the following (see Figure 2).

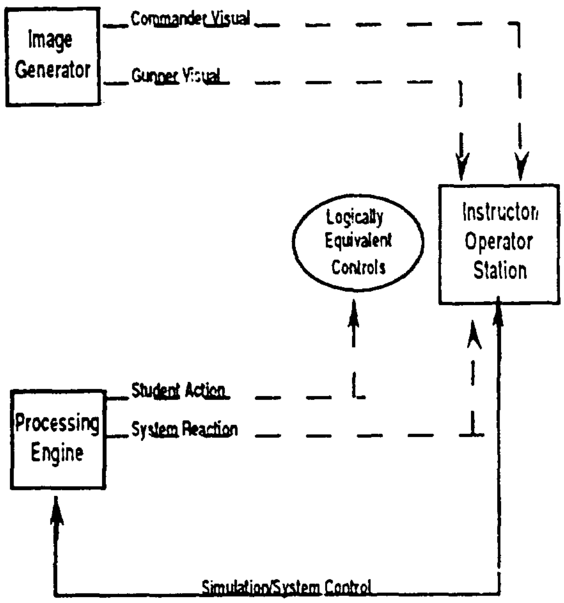


Figure 2

Starting from this perspective it has been possible, for engineering purposes, to isolate and condense the vehicle specific functionality from a crew compartment, which due to its mission as a training system is of the size and shape of approximately half a tank turret, to three or four small control boxes the approximate size and weight of a computer keyboard. These non-destructive attachments allow greater advantages than to convert an available COFT from one vehicle type into the engineering equivalent of the currently needed vehicle type. Liberated from the physical equivalence required for effective training by appreciating the sufficiency of logical equivalence for engineering development/maintenance, we can repackage our logically equivalent hardware into a more "engineer friendly" unit. Indeed, this seemingly secondary implication has, it itself proven to yield great productivity benefits. As stated earlier, the COFT systems are designed to support the interactions of three individuals: a student gunner, a student commander, and the instructor/operator. Prior to the development of these logically equivalent function boxes, development and testing was frequently a two engineer task due to the input and output functions being distributed through out the COFT system.

[1] It can not be over stressed that the use of these boxes is purely for software maintenance purposes. These boxes are no substitute for a crew station in actual training.

Now by concentrating the input and output functionality at one location, by using the output capabilities of the IOS in conjunction with logical equivalent function boxes that sit on the IOS work area, it is now possible for one engineer to perform development and testing activities that before were physically impossible. This has proven to be such a benefit, that in many cases the logical equivalent function boxes are used when no reconfiguration is needed (in a sense we are reconfiguring a COFT of vehicle type A into a COFT of vehicle type A) solely for their ability to increase ease the engineering workload and increase productivity.

RECONFIGURABLE TRAINER CASE STUDIES:

Reconfiguring an M1 Main Battle Tank Trainer (MBT) for M1A1 MBT Development.

A recently completed Software Block Upgrade required software functionality that would be deployed to both M1 and M1A1 COFTs. When hosted on the M1A1 COFT, the software was to exploit some of the unique features as found in that vehicle that are not to be found on the M1.[2] During the development of this software, a major concern was the lack of access to an M1A1 COFT for development and testing due to the fact that all such configured COFTs were fully allocated to their training mission. By applying the concept of engineering equivalence, it was possible to capture the unique features of the M1A1 and reduce it to a simple set of lights and inexpensive switches for an approximate cost of $200.00. The software developed and tested using this box was successfully deployed and is currently in operation.

[2]Due to the similarity in vehicle types, which implied that the majority of the software would be identical, it was possible in this case to actually make the software reconfigurable. Through the use of software logical names that define the vehicle type to be trained, the software reconfigures itself by the enabling, disabling or modifying certain aspects of the software logic to achieve the correct algorithmic performance of the vehicle in question. This simplified the management and future maintenance of the developed software as the software for both vehicles could be managed as one configuration item.

Using Reconfigurable Trainers Across Vehicle Families: Using a MBT COFT Trainer for Software Development for the Bradley Fighting Vehicle.

As of the writing of this paper, we are currently in the process of developing a Software Block Update for the Bradley Fighting Vehicle. This effort will require numerous software modifications and will involve several engineers working simultaneously on various code issues. We have been able to greatly increase our "hands on" development time, and thereby reduce our schedule and technical risk, by developing a set of three computer keyboard sized boxes which replicate the functionality of the Bradley controls on a logically equivalent basis and have been able to exploit available access to M1 COFT systems which are currently experiencing a lull in system usage.

LESSONS LEARNED AND THE FUTURE:

In summary, necessity, in the form of the need to perform software maintenance on members of the current family of COFT systems while constrained in the availability of these systems, has forced us to see these systems from a new perspective: logical equivalence. This perspective was derived by looking beyond the specific details of the various COFT systems which make them different, and realizing that the components that make up each system are basically the same. In fact one could say that each COFT system, whether for the M1, M1A1, or Bradley, are composed of basic objects that are instantiated for a particular vehicle. So this "new perspective" is hardly new at all, it is just an application of Object Oriented Design Methodology.

It would be wrong, however, to chalk this up as simply another case or "reinventing the wheel." It must be remembered that the current COFT systems evolved as the need for them arose and without an Object Oriented Methodology being in place. Yet, practical issues, such as logistic support and the economy of developing new systems by exploiting the features of existing systems, have resulted in a family of systems which have a strong Object Oriented flavor to them. An examination of the software for the different families of COFT trainers, the details of which are beyond the scope

of this paper, reveals that it could also be decomposed to objects and instances of these objects for the vehicle to be trained. The fact that the hardware and software systems seem to tend towards the Object Oriented Methodology, perhaps even in spite of the developers' efforts, would seem to provide a strong case for the efficiency of this methodology. Conceptually, therefore, we can visualize the creation of a COFT object, both hardware and software, which can then be instantiated for any vehicle. This approach would have much to recommend it from the software life cycle maintenance perspective. Software developed under such a concept could be managed as one configuration item with greater ease and efficiency than the current philosophy of managing the software for each vehicle type as a separate configuration item. The traditional obstacles to such an approach, processor speed limitations, memory limitations, and development costs, are disappearing as computer hardware becomes faster, bigger (in terms of memory), cheaper and development costs are more and more outweighed by maintenance costs. We could take the concept embodied in the concept of the logically equivalent reconfiguration boxes and take it to the next conceptual level: the development of a universal software maintenance system. Such a system , with vehicle specific logically equivalent attachments, could support any of the next generation of COFT systems at less cost and with greater efficiency than a target system diverted to this task. If the Object Oriented Methodology was used from the start of the process, with reconfigurability "designed in" by intent in the new generations of training systems as opposed to being derived after the fact due to necessity, it would appear that the efficiency of life cycle maintenance and management would be greatly enhanced.

SOURCE-DATA IMPERATIVES FOR CONCURRENCY

J. J. Shaw
SIMTEC, Inc.
Manassas, Virginia

ABSTRACT

The timely development, fielding and support of training systems, media, devices and courseware is critically dependent upon the quality and currency of the information (source-data) that describes the characteristics of the real-world environment for which training is required. Unfortunately, many military training programs for operators and maintainers have been, and continue to be, seriously compromised by the lack of awareness, commitment and resolve to ensure that the essential training source-data is provided as a product of the weapon system.

The solution to this problem is based on successful commercial practices and is rooted in the acquisition and systems engineering management of both weapon and training systems. The key to the solution is the implementation of structured processes that develop and maintain quality source-data products configured to both the weapon system and training system. This approach will substantially reduce the problems, risks and related costs of:

- Acquiring quality source-data in the weapon systems,
- Implementing source-data in the training systems, and
- Maintaining concurrency of the training system components.

The concurrency of the training system will be significantly improved since the required training system source-data is an integrated product development embedded in the systems and logistics engineering of the weapon system. As the weapon system design evolves, the training system source-data products will reflect the changes.

This paper provides insight into the basic source-data acquisition and implementation processes and requirements to support concurrency. Also, the paper addresses the philosophy, practices, and cost-effective methods of achieving significant improvements in source-data applicable to a variety of training programs.

ABOUT THE AUTHOR

J. J. Shaw (Jay) is the Director of Test and Evaluation at SIMTEC, Inc. He was the Program Engineer responsible for the recently completed Air Force Simulator Data Integrity Program. The principle products of that program are the Source-data process standard and handbook discussed in this paper. Jay has been involved in both military and commercial aircrew training for over 30 years, in various capacities, including engineering, program management, test and evaluation and certification of training simulators. He has worked in various industry/government working groups, and authored papers on related subjects.

SOURCE-DATA IMPERATIVES FOR CONCURRENCY

J. J. Shaw
SIMTEC, Inc.
Manassas, Virginia

INTRODUCTION

Most training system and training equipment developers are seriously handicapped by shortfalls in the information (source-data) that describes the various characteristics of the weapon system and the real-world environment. The shortfalls impact the initial development, currency and life-cycle costs of the training programs.

Training system source-data is the training oriented information that describes the real-world weapon system functionality and performance within its operating environment(s). This information is the knowledge-base for the training program and training system. Source-data is the foundation upon which the curriculum, courseware, media, training devices, associated training materials and training program decisions are developed.

The quality, timeliness and currency of the source-data is determined by the organizations that are responsible for the weapon system. In the early phases of weapon system development, the origins of source-data can be found in the engineering disciplines used in the analysis, design, manufacture, test and logistics support of the various systems and sub-systems. As the weapon system matures and is deployed, the origins of source-data are replaced and additional disciplines become involved. Unfortunately, the maturing of the weapon system does not necessarily result in improvements in the source-data.

The training system and training equipment developers find themselves in the dubious position of being critically dependent on the exclusive resources of the weapon system contractors. The dependency and associated risks are further exacerbated as the weapon system goes through the inevitable design changes. Unfortunately, in most cases, the weapon system contractors and related organizations are not motivated to provide the quality and timeliness of the source-data needed by the training organizations.

This dilemma has handicapped most military training system and training equipment acquisitions, including those programs where the prime weapon system contractor has the full responsibility for the deliverable training system. The fundamental cause is the lack of incentives for the weapon system contractors to provide the required support to the training system developers.

This paper reviews this difficult problem and discusses the recent initiatives to develop an improved approach to solving not only the initial shortfall in source-data, but also the long term concurrency implications.

SOURCE-DATA AND CONCURRENCY

One of the principle issues associated with the development, fielding and effectiveness of training systems is the proverbial problem of concurrency. Concurrency, for the purposes of this discussion, is defined as "The condition of being ready for training on the training need date."[1] In the case of emerging weapon systems, the compelling requirement is to deploy the fully developed training system at the same time that the weapon system is fielded. In the case of mature weapon systems, concurrency is driven by the objective to maintain and improve the capability of the training system components, consistent with the changes to the weapon system and the training system. Establishing and maintaining concurrency of the various components of training systems is not a trivial task and in many programs the results have been less than satisfactory.

The parallel development of an emerging weapon system and its associated training system will typically encounter concurrency

problems in the digital based systems such as avionics, weapons, flight control computers, engine controls, etc. Due to the nature and flexibility of these systems, and the typical weapon system development schedules, the change level activity becomes highly dynamic at the point in time that the training system development has reached it's crucial delivery milestones. In this environment, source-data is one of the critical factors that contribute to the success or failure of concurrency.

The relative significance of source-data with respect to concurrency was characterized in a 1989 survey of Air Force using commands. A Major representing the Tactical Air Command was asked what he considered to be the most significant components that contribute to the concurrency problem. His response was "There are two inter-related components - contracting difficulties and inadequate source-data." He further commented that each of the two components had an equal impact on concurrency.

The traditional practice of placing and accepting full responsibility for source-data on the user of the data instead of the producer of the data is fundamentally flawed in todays environment.

Further discussion in this area indicated that some of the "contracting difficulties" were, in fact, deficiencies in cost estimating attributed to the lack of weapon system engineering source-data needed to define the changes to the training system and training equipment.

Another example of this problem was indicated by an Air Force organization responsible for the acquisition of modifications for deployed training devices. This organization indicated that the modification contracts would typically involve considerable time and money just to collect the source-data needed to scope the changes to the training device(s).

Concurrency of the training system components is not exclusively effected by the design or performance of the weapon system. Changes to mission(s) of the weapon system, changes to the training environments, operational procedures, or curriculum may dictate the requirements for previously undeveloped training materials, media and revisions to training devices. Accordingly these changes in the training system will invoke the requirements for certain source-data that may not have been needed previously.

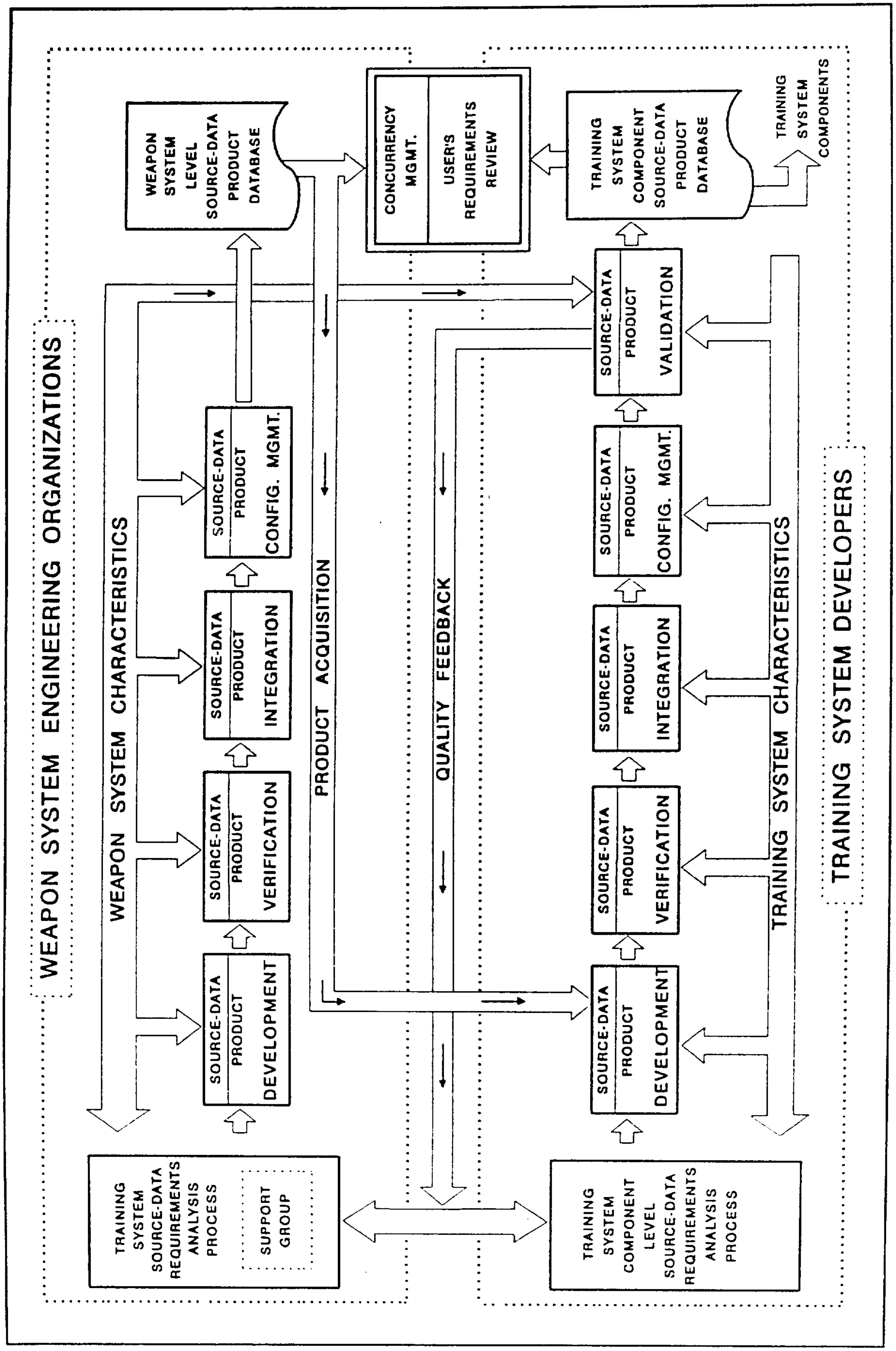
SOLUTION: DATA PRODUCTS

The recent initiative by the Air Force to establish a process for the concurrent development of training source-data as an integral part of the weapon system (and the training system) has elevated source-data to a deliverable product. Much in the same way that the weapon system contractor has been traditionally responsible for developing ground support equipment, maintenance tools, operations manuals and various other documentation, the Air Force process will develop and maintain training oriented source-data as fully configured and supported products of the overall weapon system.

The basic philosophy is to develop and maintain training oriented source-data as life-cycle products, not incidental information that may be useable in the training environment. This philosophy and the processes recommended by the Air Force are adopted from the highly successful methods used by the airline industry to develop, and maintain the source-data required for aircrew and maintenance training purposes.

The process, as described in Military Standard: Training Systems And Equipment Source-data Process, MIL-STD-XXXX (USAF), evolved from an earlier study that concluded that shopping lists for training source-data were not effective without a means of assuring quality, timeliness and concurrency of the information (products) provided.

Initially, the development of MIL-STD-XXXX focused on the objective to improve the performance, training effectiveness and delivery schedules for aircrew training simulators acquired for training programs for



SOURCE-DATA PROCESS FLOW

new, emerging weapon systems. During the preliminary fact-finding and analysis phases for the preparation of the process standard, it became increasingly obvious that the source-data problem was not isolated to aircrew training equipment. A consensus of the industry and military representatives participating in the development of the standard indicated that the process should support courseware, training materials and media for both new and existing operator and maintenance training programs. The commonality of source-data needed to support generic operator and maintainers training media and courseware requirements was examined and found to exceed 50% in most areas and above 70% in the training equipment design and performance areas. The expanded scope for source-data applications does not change the general process. It will, however, increase the requirements for the consistency, integrity, commonality and currency of source-data products.

The process described in the standard is intended to be fully integrated within the systems engineering approaches that are used to develop the weapon system and the training system. The fundamental objective is to imbed the development of source-data products within the weapon systems contractors (and sub-contractors) engineering and testing process This approach capitalizes on the irrefutable fact that the weapon system designers and evaluators are the exclusive source of the knowledge base upon which the training source-data products must be based. This approach is consistent with the "Concurrent Engineering" and "Integrated Product Development" initiatives of the DOD. In the large commercial aircraft programs, this approach has evolved during the last 10 to 15 years, and is institutionalized within the design and test engineering organizations.

SHARED RESPONSIBILITY

Traditionally, in those military programs where the training system or training equipment are procured independent of the weapon system, the training system and training equipment developers are obligated to assume the total responsibility for source-data that must be generated by other sources that have little or no vested interest in providing data. Even in some programs where the "Prime-builds-all", the training system sub-contractors are obligated to accept full responsibility for source-data that must be acquired from other organizations (including the prime) that are not motivated to provide what is needed. Typically the priorities of weapon system contractors are focused on the "big ticket" items. The associate contract agreements between the weapon system contractors and the training system/ training equipment developers cannot compete for the engineering resources. Consequently, source-data are relegated to resources that are outside the mainstream of the weapon system engineering effort. The traditional practice of placing and accepting full responsibility for source-data on the user of the data instead of the producer of the data is fundamentally flawed in today's environment.

Implementing a source-data program similar to that prescribed in MIL-STD-XXXX requires that the weapon system community accept limited responsibilities for developing and maintaining quality source-data products that are ultimately used for training the operators and maintainers of the weapon system. Likewise, the training system and training equipment developers accept certain responsibilities for their role in the overall process. The weapon system prime contractor's responsibility for the development of concurrent source-data products involves a management commitment at a sufficient level to ensure the mainstream engineering resources are applied to the effort. An example is the co-development of real time training simulation models of the weapon system dynamics as a subordinate product of the weapon system engineering simulation. A key factor is the capability of the weapon system prime contractor to effectively interface between the requirements of the training community and the weapon system engineering and test community.

PROCESS FLOW - OVERVIEW

It is not the intent of this discussion to engage in a detailed review of the process standard. However, an appreciation of the general functional flow, relationships and interfaces is necessary. The figure, Source data Process

Flow, is a generalization of the flow model that is in MIL-STD-XXXX. The upper half of the diagram depicts the weapon system process starting with the Source-data Requirements Analysis. The weapon system level source-data products are maintained in the database, and acquired by the training system developers. The lower half of the diagram contains the training system source-data process, also starting with the training system component level source-data requirements analysis. The required weapon system source-data products form the foundation for the development of the source-data products to be used in the training system process. The source-data product validation process takes place in the training environment. The validated products flow into the training system component level source-data product database. In addition, a quality feedback is provided to support improvements and integrity of the products. The developers and maintainers of the training system components acquire the source-data products from the database. The information that resides in both databases is reviewed by the training system users as part of the concurrency management process.

The key to providing concurrent source-data products is embedded in the configuration management process of both the weapon system and the training system. In its simplest form, the source-data products developed by the weapon system contractors are weapon system configured items (CIs). As the weapon system design matures and is changed, corresponding changes are required to those source-data products effected by the modification. This applies throughout the life of the weapon system. Also, as indicated in the flow diagram, the contents of the source-data products is determined by the characteristics of the training system. For example, if the maintenance training curriculum requires that additional hydraulic failure modes be incorporated in the training courseware, then the requirements and specifications for the designated source-data product(s) will be revised. Likewise, if the hydraulic characteristics of the weapon system are changed, then the configuration of the source-data product is affected.

Within the training system, configuration management of source-data products is allocated to the appropriate training system components, i.e., training programs, courseware, CBT, Training Devices, etc. Changes to the weapon system source-data products that will impact the currency of the training components are tracked continuously through advisories generated by the weapon system contractors. Accordingly, as the configuration of the training system is changed, corresponding advisories are provided to the cognizant weapon system offices.

Ultimately, under ideal conditions, these processes will establish and maintain a database of verified and validated source-data products that will:

- be consistent with the configurations of the weapon system and training system,
- conform to the quality requirements of the training system/equipment developers and operators, and
- support the decision making process of the training system organizations for acquisition, development and modification of the training system components.

The overall process prescribed by MIL-STD-XXXX is a quality based continuum that begins as early as possible in the life of the weapon system and continues throughout the life cycle. The training and weapon system developers, operators and life-cycle support groups establish and maintain the interface(s) required to meet their mutual objectives. Likewise, the relationships between prime and subcontractors for both the weapon system and training system should be structured for the long term life-cycle requirements. As the programs evolve, the quality, concurrency and integrity of the source data-products will be substantiated through verification and validation as applied in the training environment.

GOVERNMENT ROLE

To effectively implement a program of this type in the military environment, it is imperative that weapon system program offices establish the appropriate priorities required to ensure the ultimate success for concurrent development of source-data products. The Systems Engineering Management Plan (SEMP) for the weapon system should include the MIL-STD-XXXX process for the development of source-data products at a high enough level to sustain the inevitable pressures from competing program elements.

The key commitment to be extracted from the weapon system prime contractor is the establishment of a "training-smart" management position responsible for the development of quality, concurrent source-data products. This position should have adequate clout to ensure that the engineering resources within the main stream of the weapon system program are properly applied to the source-data objectives. Also, the commitment should obligate the weapon system sub-contractors to provide the same level of engineering support. Last, but not least, is the commitment of the weapon system contractors to ensure the verification of the source-data products.

The most critical step in the process of developing source-data products is the initial source-data requirements analysis, which translates the training system needs into weapon system product specifications. To aid in the requirements analysis process, the Air Force has developed a military handbook[2] that provides guidance for the identification of source-data requirements. The government program offices for the weapon system and training system components should cultivate a highly motivated Source-data Requirements Analysis Support Group that represents not only the weapon system and training system contractors but the military using command. The value of this working group should not be under-estimated. Properly supported by the government, they will provide cost-effective solutions for the constraints and limitations placed on the weapon system and training system. As a practical matter, the working group should serve to reduce the cost of source-data development by maintaining cost/benefit objectives.

The source-data requirements analysis is an iterative process that continues throughout the life-cycle of the weapon system. The source-data requirements support group is the catalyst for maintaining the quality, timeliness and, above all, the training objectivity of the overall effort. If the interests of the ultimate end-users, i.e, the deployed training system organizations, are not adequately represented in the support group, the effectiveness will be seriously compromised.

The acquisition planning for source-data products must support the short term implementation requirements and the critical long-term, life-cycle support for the weapon system and training system. In today's environment some weapon system platforms and associated training components are exceeding 35 years of operation. During the life span of these systems, major changes in functionality, mission and operating requirements have occurred. The lessons-learned are that the development of second and third generation of training system components is seriously compromised by the inadequate and poorly supported source-data. The long term planning factors should include the potential transfer of weapon system engineering responsibility to other organizations such as military logistics support depots and contractors other than the original developer. Likewise, the long-term support of the training system components may be absorbed in other training organizations. The long-term planning should ensure the integrity of the source-data products, processes and interfaces required to support concurrency.

MECHANIZING THE PROCESS

The overall implementation and control of the source-data process is rooted in the information management systems that are widely employed in both the commercial and military environments. Once the initial training requirements are effectively translated into real-world functional characteristics, and the appropriate resources are committed, the

remaining steps are commonly used in weapon systems engineering. This structure is intended to be tailored by the weapon system organization to achieve the most cost effective integrated product development that will support the quality and concurrency objectives. The co-development of the weapon system and source-data products within the same engineering groups will ensure the highest level of commonality. As the weapon system design matures, the corresponding source-data products will be developed, integrated, verified, and placed under configuration control. This process establishes the essential foundation for the initial development and life-cycle concurrency of the training system. The concurrency of the source-data products is an extension of the configuration control of the weapon system and the training system.

The source-data products in the weapon system database represent the baseline configuration of the integrated weapon system. These weapon system source-data products are selected and acquired by the training system developers.

The training system developers will then produce component level source-data products that are derived from the weapon system baseline. Integral to this effort is the source-data validation process accomplished in the training system environment. The validation process also provides a quality feedback to the source-data requirements process, so that defects in source-data are identified and appropriate changes are made to improve the source-data products. The source-data products in the training system database represent the baseline configuration for each of the training system components.

The composite of the two databases provides the critical information needed for the training system users reviews to support the concurrency management decision process, planning, acquisition, development, quality, and currency, of the training system.

TAILORING THE PROCESS

The basic process as described in MIL-STD-XXXX is designed to be tailored for a variety of applications. The core elements as illustrated in Figure 1. can be adapted to most military applications ranging from the primary training programs to the emerging advanced tactical weapon systems. The tailoring objectives must focus on the:

- Training system requirements (short and long term),
- Weapon system program requirements, and
- Weapon system contractor's approach including:
  - Subcontracting
  - Engineering, and
  - Testing

The most demanding requirement is the effective integration of the source-data product development process within the mainstream of the weapon system contractor's systems engineering and testing organizations. The determination of what engineering group, or groups, are best suited to support the development of source-data is driven by the level and type of information that is identified in the source-data requirements analysis. For example, in the case of source-data needed to develop training simulation models, it would be appropriate to assign the responsibility to the engineering simulation resources.

In most cases, the tailoring or adaptation of the process will be driven by the weapon system development and test schedules. The limited windows of opportunity to collect certain critical data will influence the process and resources. The tailoring process involves the various sub-contractor's engineering and test organizations. These factors, and others, should be taken into consideration as part of the source-data requirements analysis.

In the case of previously developed weapon systems or platforms, the tailoring and adaptation may involve the use of resources and methods that would not otherwise be cost effective. For example, if the flight test program for a previously developed aircraft will not cover the areas of the performance envelope needed to support training simulation, then alternative data collection methods should

be considered. In this case, the overall process including the source-data requirements analysis should accommodate these alternatives.

CONCLUSIONS

Military training system developers are reliant on the resources of weapon system organizations for the quality and timeliness of the source-data that describes the performance and functional characteristics of the weapon system. The lessons learned and the successes of the airline industry are the foundation for the Air Force initiative to establish the processes required to develop and maintain quality source-data as configured products of the weapon system and the training system. The processes are imbedded in the mainstream of systems engineering and integrated product development of the weapon system and training system. These processes will provide the database necessary to support concurrency of the training system.

To be effective, the government must provide the program incentives necessary to implement and institutionalize the source-data process. The weapon system contractors and sub-contractors will provide the source-data products and associated services if the requirements are reasonably definitive and integrated within the overall program requirements.

The responsibility for the quality and timeliness of the source-data products is properly placed and shared by the weapon system and training system organizations. The end-result is substantially improved concurrency, quality and cost-effective training systems.

------- REFERENCES -----

1. Guide for Management of Air Force Training Systems - AF Pamphlet 50-11 (DRAFT), 2 April, 1992, Glossary of Terms.

2. Military Handbook, Aircrew Training Equipment Source-data Requirements - MIL-HDBK-ATESR (DRAFT) 14 September, 1992.

USING COGNITIVE SIMULATIONS IN MULTIMEDIA FOR MAINTENANCE TROUBLESHOOTING TRAINING: PRACTICAL, COST-EFFECTIVE SIMULATIONS

T. Kent Thomas
Allen Communication, Inc.
5225 Wiley Post Way
Salt Lake City, Utah 84116

ABSTRACT

Experts appear to master the art of critical thinking in troubleshooting. It's as if they have a mental model of the system etched on the inside of their forehead. How can this mental model be transferred to the novice? Through carefully crafted multimedia courseware and free-play simulations, novices can match wits with the expert in a delivery environment that doesn't require either expensive expert systems software development nor complex hardware simulators. Preliminary training results from a 200-plus hour program suggest that interactive multimedia courseware may produce results approaching both of those methods, with substantially lower development and delivery costs. Small-group tryout results from 21 courses developed by Allen Communication for Air Force maintenance technicians show a 25% aggregate increase in knowledge, and a striking 79% aggregate leap in the ability to successfully apply expert troubleshooting strategies to simulated problems.

The mental models of experts, the sequence of troubleshooting actions they perform, and their reasoning have been captured using cognitive task analysis methods and used as the basis of courseware design. Experts' mental models form the foundation of the tutorials that comprise approximately 70% of the courseware. Their performance on complex troubleshooting problems is the basis of the simulated troubleshooting scenarios. Combining this detailed cognitive task analysis with high-impact motivational video, focused in-depth tutorials that directly depict the mental models of experts, and extensive free-play simulations, this F-15/F-16 Maintenance Continuation Training Program won the 1993 *Nebraska Interactive Media Award* for the most significant achievement in the Government/Military category and an Intermedia *Invision* Bronze Medal.

The author will present an overview of the methods used to design and develop these simulation-focused multimedia courses, including: knowledge engineering, design, programming, and evaluation. Courseware samples will be demonstrated and preliminary results reported.

ABOUT THE AUTHOR

Mr. Thomas is a former Air Force education/training officer and computer systems requirements analyst. He has a B.A. from Carson-Newman College and is completing a Masters in Instructional Technology from Utah State University. While previously on active duty, he planned and managed a variety of new applications of training technology, including the design and implementation of Air Education and Training Command's largest operational computer-managed instruction system and Air Combat Command's maintenance continuation training program using interactive videodisc. He has managed the design and development of over 70 multimedia courses, including four winners of the *Nebraska Interactive Media Award* (three consecutive winners in the government/military category) and three winners of the Intermedia *Invision Award*. A frequent presenter at ITTE/IITSEC, Mr. Thomas is Vice President, Research and Development, at Allen Communication and a Major in the Air Force Reserve.

USING COGNITIVE SIMULATIONS IN MULTIMEDIA FOR MAINTENANCE TROUBLESHOOTING TRAINING: PRACTICAL, COST-EFFECTIVE SIMULATIONS

T. Kent Thomas
Allen Communication, Inc.
5225 Wiley Post Way
Salt Lake City, Utah 84116

INTRODUCTION

Air Combat Command (ACC) has been using interactive videodisc courseware for several years to provide maintenance continuation training (Thomas, 1987). Although quite effective, this courseware has primarily focused on systems knowledge and procedural training to support weapons systems conversions such as to the F-16 Block 40 and F-15E. Consequently, the maintenance problems related to troubleshooting these complex, highly integrated systems have persisted. Further, this courseware for conversion training has been based primarily on the design principles of behavioral psychology (Hannafin and Rieber, 1990), with either weak or nonexistent simulation capabilities. Meanwhile, ACC has been a strong supporter of research programs that evaluate technologies (such as "cognitive task analysis" and "simulation-based intelligent tutors") that address these persistent problems -- especially the ongoing Basic Job Skills (BJS) project conducted by Armstrong Laboratories. This paper documents the early results of transitioning some of this research to full scale development in the F-15/F-16 Maintenance Continuation Training Program (MCTP). It describes both lessons learned from cognitive-based courseware design and indicates areas with high potential for further research.

According to the Productivity Investment Funding (PIF) package that supported the MCTP, approximately one quarter of the components removed from the F-15 and F-16 during flight line maintenance are actually serviceable (F-15, 25%; F-16, 27%) when bench checked. While there are many factors other than training (or the lack of it) that directly contribute to these statistics (including management pressures for "quick fixes"), poor troubleshooting skills are obviously a part of the problem. These challenges are further compounded by initiatives such as Rivet Workforce (which tripled the technicians' responsibilities in some cases) and two-level maintenance (which provides no local "back-shop" support to test and verify the condition of the components removed during maintenance). These troubleshooting weaknesses are expensive in terms of time, logistics funds, and their potential impact on readiness. The potential paybacks identified on this \$4.2M courseware investment are \$11M in the first year and over \$100M in savings across the anticipated life-cycle of these weapons systems.

PROGRAM DEVELOPMENT GOALS AND REQUIREMENTS

The F-15/F-16 MCTP contract required the development of 21 courses (10 for the F-15 and 11 for the F-16), each approximately 10 hours of average student contact time, which addressed ten separate maintenance specialties. It represented a ground-breaking sylthesis of several isolated Air Force research and development projects. These innovations, when compared to other interactive courseware development efforts, lie in four primary areas: Knowledge Assessment Tools (KATs), Cognitive Task Analysis, Simulation-Based Instruction, and Automated Analysis Package.

Knowledge Assessment Tools (KATs)

Drawing upon previous Air Force research in assessing the *knowledge and skills* of technicians on the A-10 Stability Augmentation System and the F-15 APG-63 Radar System, the MCTP contract required the development and validation of tests prior to designing the associated course. This approach formalizes and implements the concept of "test before you train." If a technician does not need the training on a given topic or system, the test results allow the student to be "routed around" that particular portion of the courseware. This was done to significantly improve the training efficiency and the students' motivation since the students were to be trained only where needed.

Validating the KATs early in the development process (in

the analysis phase, prior to design) allows the pretests to be much smaller and more efficient. Rather than a comprehensive test of all the content included in a given course or module, only key questions or problems that have been statistically validated to discriminate between novices and experts are posed to the student. This further reduces the amount of time the students are tested and trained. Validating the tests and analyzing the data prior to design eliminated many potential topics from the training -- topics whose data did not support their inclusion since the content was commonly known -- while other topics were either included or their coverage expanded as a result of this analysis. This process ensured that the training focused on only the most needed areas. Finally, much invaluable data about common misconceptions and misunderstandings was gathered for inclusion in the later design of the associated courses.

Cognitive Task Analysis

Cognitive task analysis is a set of procedures to define the differences between how novices and experts think about a given system or problem. As previously mentioned, ACC staff personnel have been strong supporters of the Basic Job Skills (BJS) program which researches the use of a cognitive task analysis methodology to design "intelligent tutors" for teaching complex troubleshooting performance. By focusing on the experts' thought processes and presenting them in the context of robust training simulations, preliminary indicators from the initial test of the BJS F-15 Avionics Manual Test Station Tutor revealed a significant potential to condense the on-the-job troubleshooting experience into a short training course (Gott, 1989).

Though the MCTP contract did not require a specific cognitive task analysis methodology, it did require that it be performed in the analysis phase, prior to design, and that the results be used as the basis of designing the courseware and, specifically, the troubleshooting simulations. It clearly stated that the goal was to teach novices to model the experts' performance on troubleshooting problems -- i.e. to teach them to think like experts.

Simulation-Based Instruction

Acknowledging the importance of simulations to teach complex tasks related to complex systems, the MCTP contract required approximately 30% of the courseware to be simulation-based. Simulations are generally regarded as necessary in order to provide relevance to "real-world" tasks, yielding more valid assessment of knowledge and skills while enhancing the transfer of learning to the actual job environment. Further, the MCTP required that these simulations be based on the results of the cognitive task analysis as previously discussed.

While this emphasis on simulations incorporates some of the research findings from the BJS project, note that this approach differs significantly from that used in the actual design of the BJS tutors. The BJS tutors use only simulations with "coaching" as an instructional strategy -- all learning takes place in the context of a simulation, with "hints" or "help" available upon the student's request. The MCTP contract allowed approximately 70% of each course to be provided by other instructional strategies, primarily the more common (and less expensive) tutorials.

Automated Analysis Package

Intended for use by senior managers and staff, the MCTP contract required the development of automated programs so that planners and managers would have the necessary data to show both the training effectiveness and cost-effectiveness of the program. Specifically, it required automated programs that correlated pretest to post-test scores by course and module to show training effectiveness of each course. It also required pretest versus post-test correlation by core maintenance tasks (from the Specialty Training Standard or Job Qualification Standard) and by the work unit codes reported in the Core Automated Maintenance System (CAMS) to show cost-effectiveness. In a nutshell, it required automated programs to point toward maintenance trends on those troublesome systems that would "prove" the program met its goals. Much of the data reported here came from those programs.

METHODS USED

The original timelines specified in the MCTP contract reflected an intense development cycle, given all the data requirements, analysis, and reviews. The first four courses were to be delivered within 10 months of contract award and the remaining 17 on a 12-month development cycle. All were to be delivered within 32 months. As the first four courses were under development, it became evident that these time frames

were too stringent. Changes to the requirements were negotiated. The development cycle was extended to 14 months for each course (at no additional cost to the government), though overall program duration remained fixed. The KAT development and cognitive task analysis schedules were merged to allow them to occur concurrently rather than sequentially, as specified, yet still prior to courseware design.

Further, this effort was ongoing during the *entire* duration of Desert Shield and Desert Storm. Due to high reliance of this project on Air Force personnel and support, the impact of this large-scale military operation was all-pervasive. Schedule changes and personnel changes were frequent, as were limitations on reviewers, sample students, and access to equipment or systems. The design and development methods used were under constant refinement, change, and full of "work-arounds." The following discussion of the methods used is provided as a *general,* somewhat "ideal" process, that likely was never fully incorporated on any one course.

KATs and Cognitive Task Analysis

As discussed earlier, the goal of the KAT was to develop a diagnostic device that was "better, faster, and cheaper" than a traditional pretest. It must sample learner knowledge and skill in critical areas, assess current level of performance, and direct the learner to the appropriate instruction. The following development process was used.

1. Screen and identify project subject matter experts (i.e. SMEs). A critical part of any cognitive task analysis or "knowledge engineering" effort is identifying *the right SMEs.* Since their understanding of the subject system is to be captured as the basis of the software system to be developed, they must truly be experts. A variety of methods was used to identify SMEs, including: 1) surveys for screening subject-matter experts and review their experience, 2) interviews with SMEs, managers, and supervisors, and 3) perhaps most valuable, soliciting recommendations from their peers. Though the peer recommendations were almost invariably more accurate and were relied upon more, they were not used without corroboration by other means.

Questions were posed to these members of the target audience such as "If you were faced with a very difficult problem on system XXX that you have been unable to solve, who would you choose to help you solve it (from any base in the Air Force) and why?" Several interesting points came to light in this process, including: 1) the peer groups understand and acknowledge expertise in troubleshooting, 2) individual expertise (even for those acknowledged as "experts") varies widely from one system to another system or to another part of the same system, and 3) there are very few, if any, experts on entire systems -- their expertise is *very domain specific.* Frequently the experts on one subsystem were no more than average performers on a related subsystem.

2. Develop knowledge surveys using initial SMEs. Usually two SMEs identified by the above process were employed to develop initial knowledge surveys used to further screen their peers. Note that this survey was not intended to truly determine expertise, only to place the individuals in an initial category for further analysis. Note also that the questions included in the survey were frequently word problems that described a troubleshooting scenario or very technical questions about the specific functioning of a system or subsystem. At the same time, these SMEs were asked to define representative troubleshooting problems (including problem set-ups and step-by-step solutions) for possible inclusion in the course. A preliminary cognitive task analysis was performed for these problems using the PARI method (described below).

3. Perform cognitive task analysis of both novices and experts. Developed as part of the BJS program, the PARI methodology is a set of data collection procedures, including structured interviews, designed to capture and document an expert's actions and reasoning processes simultaneously (Hall et al, 1990). Each step in the expert's performance is analyzed to determine the Precursor (i.e. pre-existing condition), their specific Action and the reasons for it, the associated Results of that action, and their Interpretation of those results, hence the acronym PARI. A somewhat streamlined version of the PARI method was used to interview approximately four experts and four novices from at least two different Air Force bases. The changes to the published PARI methodology were minor, intended to support "mass production" and (in the author's opinion) do not substantively change the results.

Note three minor, yet perhaps valuable, changes to the PARI technique. First, the term "problem" was used in lieu of "precursor" since it communicates better to laymen and since the troubleshooting strategy to be

taught was space-splitting (also known as split-half or half-split). This change in terminology kept both the interviewers and interviewees focused on the "problem space" and how each step did or did not continue to reduce it. Note that each of the courses presents the PARI technique directly to the students, in the context of a "generic" Troubleshooting Techniques lesson, and again in the context of a guided, system-specific simulation, using this terminology of Problem-Action-Result-Interpretation.

PARI interviews are very dependent on visual representations of a mental model of the system. The second change was to provide the "experts" with drawings of the mental models developed by the initial SMEs, rather than having them draw their own for illustrating their solutions. They could critique or modify them as necessary. The goal was to develop and document a common, *fundamental mental model* (Lesgold et al, 1988) that could be incorporated directly in the courseware to illustrate the troubleshooting process to the learners, and to simultaneously document and depict how it is expanded, modified, or otherwise manipulated as the troubleshooter delves deeper into the problem. While this technique generated considerable discussion between the SMEs, consensus could almost always be gained. This resulted in a common, yet robust, mental model of the system (though perhaps not an optimum nor universal one) that can be validated for functionality and illustrated via computer graphics and animation in the courseware. [See Wilson and Rutherford (1989) for a further discussion of mental models.]

Finally, the interviewing of novices was treated almost as critically as the interviewing of experts. The mistakes and misunderstandings of novices were documented as clearly as possible since common mistakes and misconceptions were to be directly addressed in the courseware. The combination of actions taken at each point by both novices and experts would define the limits of free-play to be designed into the simulations.

4. Develop alpha version of KAT and review it with SMEs. Based on the cognitive task analysis performed above, the knowledge survey was expanded to include more "key tests or checks" and the interpretation of results, in addition to the word problems and technical questions previously developed. This is the foundation of the KAT. The KAT items were then reviewed by the initial SMEs for accuracy prior to field testing.

5. Develop beta version of KAT and validate with field SMEs and novices. After any necessary revisions, the beta versions were administered to representative members of the target audience at two or more Air Force bases. The sample size was ideally 20 members, with a fully representative range of experience on this specific weapons system. More subjective data about the subjects was gathered simultaneously, including evaluations of their performance from their supervisor(s) and a summary of their background and experience.

6. Analyze results and select final items. Responses to KAT items were analyzed, item by item, in three different ways using a point-biserial method to determine item validity. In fundamental terms, this validation identified KAT questions that discriminate between novices and experts. Subjects were first categorized for analysis by a subjective evaluation (based on experience, and supervisor and peer input) as experts, average performers, and novices. The central category was added to accommodate those who are truly not experts nor novices. Then, the subjects were categorized by the skill level from their Air Force Specialty Code (AFSC), i.e. 3-, 5-, and 7-level. Each item was evaluated for significant differences between the results for the subjective grouping and the subject's AFSC skill level. Finally, each student's overall KAT score was used to assign him/her to the subjective grouping and the items analyzed to provide an internal referent. Minor or occasional anomalies were expected and accepted. By definition, any test item with a positive point biserial value will discriminate. However, only the items with the highest positive values were included in the courseware, since the higher the value the better the item discriminates and the more confidence that can be placed in the results. Though there were significant variations across all 21 KATs, the vast majority of item validity scores were above 0.33. The KR20 reliability index was calculated for each test to evaluate its reliability. By its simplest definition, reliability is the ability of the test to provide consistent results, all external factors being equal. A composite threshold value of 0.80 was established for the reliability of each test, considered as a whole. However, given the sample sizes for KAT validation efforts, this should still be considered a rough estimate of reliability.

Given the administration of the KAT to two pools of students at two bases and the fairly high validity and

reliability criteria used, there was confidence that the KAT would make consistent, accurate diagnoses of knowledge. The final step was to refine the objectives for each course module to be developed, correlate the KAT items to the objectives, and ensure adequate numbers of valid and reliable items were available for content coverage. In those isolated cases where adequate, validated items were not available, additional items were generated and validated in conjunction with small group tryouts.

Simulation-Based Instruction

While the simulations are the "heart" of the instruction, they are only a part of the overall design. The KAT and cognitive task analysis results are incorporated throughout each course, as is careful attention to the target audience.

General Structure and Design. Since the target audience for this training includes both apprentice and journeyman maintenance technicians, and the training is performed on a time-available basis, modularity is critical. Each course is designed so that students may enter quickly, access their previous position in the course, receive intensive instruction, and then exit easily. Overviews, introductions, and navigational lessons are mandatory for first-time students and then available as options. The courseware allows the student access to various options such as Help, Take a Break, Course Status, Expert Tips, and flight line Logbooks. Advance organizers and summaries orient students to the overall content of the modules. Students can exit the course quickly by selecting a Logoff option, which places a "bookmark" allowing them to reenter the course at the point they left it. Bookmarking also allows the students to browse through and repeat instruction they have already seen. Each course uses the same structure and conventions, since one student may take several courses in the MCTP series.

Motivational and affective learning strategies were integrated to make the instruction appealing to the students and encourage knowledge transfer. For example, a theme song was commissioned to a professional song-writer and mixed with high-impact video (including considerable Desert Storm footage). It builds and reinforces the intrinsic motivational theme of *"When The World Has Its Eyes On Me"* and is incorporated throughout each course. On-camera role models introduce, guide the student, and conclude each module. Learning games similar to *Concentration*(tm) or roulette wheels (where students can wager points against their ability to answer technical questions) are used for review, practice, and reinforcement. "Commercial breaks" are interjected throughout to surprise and entertain the student, helping to maintain motivation.

The instruction in each course consists of four modules covering: Subject System, Related Systems (and their interfaces with the subject system), Diagnostic Tools and Procedures, and Troubleshooting. The troubleshooting module is simulation-based, as discussed earlier, while the other modules are largely tutorials, with linear introductions and gaming strategies for reviews and practice. Test items from the KAT development process are included as separate pretests and progress checks for each of the first three modules. Students enter the troubleshooting instruction and free-play simulations only after passing KATs or completing the previous modules. If the KAT determines that the students need instruction, they are routed to the appropriate module (which contains two or more lessons). Each lesson instructs the student about a system's main components, functions, working relationships, and common troubleshooting problems. These "chunks" of information, over 30 per course, are called clusters.

Each cluster further evaluates the student's knowledge and ability via a conversation-like Socratic dialogue technique, and then, depending on the results, will route the student to in-depth tutorials. This dialogue provides a more discrete "diagnosis" once the KAT has indicated that a weakness exists. This design allows the more knowledgeable students to progress very quickly through the course by correctly answering questions. Weaker students receive training that is individualized to address their specific weaknesses. This approach has allowed some students to complete a course in four hours or less, while others spent the two days allocated for tryouts and still were unable to complete the course.

In these tutorials, students receive extensive training about the system which fills in their specific "knowledge gaps." The instruction in each lesson is accomplished by using voice-over narration or text, supported by full-color graphics and video of proper maintenance procedures. The mental model for the system is introduced in the Subject System module and is explained in the context of describing the system's functionality. It is reinforced in the Related Systems module, where it depicts the interface(s) to the related systems on the aircraft. In the Diagnostics module, it is

used to illustrate where certain tests are performed, test equipment is used, etc. It is also incorporated in the guided simulation of the Troubleshooting Module as feedback to depict and reinforce effective space-splitting.

Troubleshooting Simulations. Each course contains six to nine specific troubleshooting simulations, used throughout the final module. Simulations are used as the pretest for the module, the specific lessons within it (a generic Troubleshooting Techniques simulation as discussed earlier, plus a course-specific guided simulation), to provide practice in application, and they serve as the progress check. These are fairly robust and powerful free-play simulations. The student can perform between 60 and about 150 actions (depending on the course) on the system in question, in any sequence at any time. These actions are as varied as talking to the pilot, researching the maintenance history, performing tests with diagnostic equipment, making visual inspections, replacing parts, and performing operational checks. Each simulation contains well over 1,000,000 possible paths, and some significantly more.

All the while, performance indicators track and display a running evaluation of the student's performance. Each step is evaluated to meet one of four criteria: 1) critical to successfully solving the problem, 2) a reasonable step, though usually neutral in value toward solving the problem, 3) an unreasonable step, or costly in terms of time or parts, or 4) a safety violation, which results in immediate termination of the simulation. The total number of steps taken is displayed, as is the number of critical steps and unreasonable steps, the time expended (in terms of the actual maintenance time it would take to perform the action), and the cost of parts used. Students are allowed to take three unreasonable steps before the simulation is terminated to provide feedback and a summary evaluation. The simulation is also terminated if one or more of the efficient time, total steps, or cost limits (140% of optimum) is exceeded. In the summary, the student's performance is directly compared and contrasted with the expert's performance in terms of all these factors. A step-by-step comparison of each action taken and the corresponding result is displayed. Students must successfully troubleshoot at least two randomly generated aircraft system faults before passing the course.

[illegible] being instructionally powerful to depict [illegible] relationships in a complex system, these simulations were consciously designed and used to provide the most value for the least cost to the government. While definitely effective, simulation-based training is expensive -- interactive courseware simulations can often cost at least three times as much to develop as simple interactive tutorials. Yet, the relative training effectiveness of simulations versus other instructional strategies is not clear (Fletcher, 1990). The MCTP attempts to use a more optimum mix of simulations with other strategies to increase *cost-effectiveness*. Instead of all the content being presented in the context of simulations (as do most "intelligent tutors" for maintenance skills), simulations are used here almost exclusively for illustration, practice application, and assessment. Tutorials that are focused on the carefully-analyzed knowledge and cognitive abilities that support troubleshooting are used as the primary presentation strategy.

Further, the courseware uses direct instruction methods to impart the information in the most time-efficient manner. Rather than leaving the students to formulate their own mental model of the system, a common one is presented directly. Rather than inferring the expert's approach, their specific troubleshooting sequence is presented directly for step-by-step comparison with the student. Rather than implying the cost and/or value associated with each step or action, they are displayed constantly as running totals in the performance indicators, and again in a detailed step-by-step summary after each simulation.

What is simulated also differs from most instructional simulations (Alessi and Trollip, 1991). Rather than building an extensive, realistic simulation of the system under study, likely human actions are modeled. The actions that novices and experts are likely to take are incorporated as limits to the free-play choices. The realism (or fidelity) was also focused on the areas perceived to have the most instructional value. While the results of a student's actions are clearly depicted with pictures (and sometimes sound or explanatory text) to support the student's interpretation, quite often the action choices are merely "menu selection items." Similarly, these simulation scenarios are used fully and thoroughly. A typical student will experience almost every simulation scenario in each course, even though they are presented randomly (with extinction, then "reshuffled") to prevent compromise. In comparison, a typical computer-driven flat-panel avionics trainer may provide the capability to simulate over 100 problems, yet

the student will only experience about ten of these in a typical course, due to the limitations on training time and course length.

Also impacting cost-effectiveness over the life cycle of training programs is the way the simulations (and all the courseware) were developed with a commercial off-the-shelf (COTS) authoring system. They were developed entirely in the *Quest* (tm) Multimedia Authoring System, as required by the MCTP contract. Consequently, these simulations can be updated and maintained by experienced Air Force subject-matter personnel, while neither intelligent tutors nor flat-panel trainers can be. Updates can also be affordably done via competitive contract if necessary. The simulations are "driven" by a small, yet very powerful "simulation engine" written in the underlying *Quest* (tm) Authoring Language that is compiled for speed and used for all the courses. The screens used to present specific actions and results are in lesson files, with no inherent branching or paths, while all logic and scoring criteria are contained in a delimited text file (i.e. table) that is created and updated using a spreadsheet. The "simulation engine" reads the action taken, determines the appropriate result based upon the rules table, and displays the correct result screen. This modular approach, separating logic from content, speeds up initial development and makes the simulations much easier to update and maintain, reducing life cycle costs even more.

Automated Analysis Package

All these innovations may be moot without the final one -- an automated analysis package that links learning gains back to the maintenance environment. The previous innovations were transitioned from research and development (R&D) to large-scale implementation. This last one is still an R&D effort -- to the author's knowledge, nothing like it has been attempted in either the military or industry.

Pretest and progress check scores for each question or simulation scenario are captured for each individual in each course. This data is then loaded to a *dBase III* (tm) compatible database format for consolidation and further analysis by specialized programs. These programs provide correlation of these scores by module and course to show training effectiveness data, a common practice. More importantly, they provide a [illegible] of test questions and scores [illegible] work unit codes, as discussed earlier.

Summaries are provided by student, an aircraft maintenance unit, a fighter wing, an Air Force Specialty (AFS) or an aggregate of all students who have taken a given course (regardless of AFSC or unit of assignment). This same test item data can also be loaded to an Air Force-developed test-item analysis package for evaluating individual test items and tests in terms of difficulty, reliability, validity, etc.

In addition, critique data from a standardized, Air Force-provided 17-question automated survey is gathered from each student for analysis. There is also a vehicle for unsolicited feedback in a "free-form, anonymous mode." The results described below were gathered using these standardized packages.

RESULTS

The F-15/F-16 MCTP appears to have successfully transferred research and development tools and techniques to a "production line." All 21 courses have been delivered and accepted by the government. The data available at this point is from small group tryouts of each course, since some of the courses have just been delivered and several have yet to be fielded. More meaningful data should become available when each course receives a scheduled follow-up data extraction and analysis. By the time this paper is published, that data should start becoming available.

Limitations of Available Data

Given the source, the sample size per course is small. Conclusions would be premature for any specific course. Note also that the skill distribution of students at small group tryouts was not truly representative of the target audience. An equal representation of novices, average performers, and experts was sought for small group tryouts. However, since experts could logically be expected to score highest on pretests, yet make up the smallest percentage of the target population, the data from a numerically representative sample should show even higher gains in knowledge or skills. The data has not been separately analyzed by skill level.

Another limitation is the varying numbers of students who completed a given module, especially the final one, Troubleshooting. Much of this is due to the inability of students to complete the course in two days (the maximum time available). There were also a few cases of power interruptions and hardware or software failure

during tryouts which impacted limited portions of student data. These inconsistencies have been accommodated and compensated for in the analysis of results where possible. For example, only those completing a progress check are calculated in the pretest data. Though the available data is *preliminary* by definition, coming from small group tryouts, each course successfully met the tryout criteria specified by the Air Force. When considered in a holistic fashion, some indications should be valid related to the effectiveness of the basic approach and the courseware design, due to the commonality across all courses and the aggregate sample size of all tryouts.

Table 1. MCTP Knowledge Gains -- Pretest vs. Progress Check Increases (Data From Small Group Tryouts)

Course	Subject Systems Module Pretest	Subject Systems Module Prog Ck	Subject Systems Module Difference	Related Systems Module Pretest	Related Systems Module Prog Ck	Related Systems Module Difference	Diagnostics Module Pretest	Diagnostics Module Prog Ck	Diagnostics Module Difference	Total Difference
F-15/8	50	89	39%	NA	NA	NA	51	94	43%	41%
F-16/3	53	88	35%	54	96	43%	58	90	32%	37%
F-15/2	42	84	42%	59	98	39%	69	98	28%	36%
F-16/9	48	86	38%	55	92	37%	56	84	28%	34%
F-16/1	56	83	26%	58	89	31%	46	88	43%	33%
F-16/10	63	92	28%	59	92	33%	60	83	23%	28%
F-16/11	53	85	31%	63	86	23%	61	90	29%	28%
F-16/7	65	91	25%	58	89	31%	69	93	25%	27%
F-15/5	72	82	10%	68	100	32%	70	100	30%	24%
F-15/10	68	91	23%	69	88	19%	62	91	29%	24%
F-16/4	64	86	22%	61	81	20%	57	86	29%	24%
F-16/6	72	88	21%	56	88	31%	65	85	18%	23%
F-15/1	59	87	29%	62	96	34%	83	88	5%	23%
F-16/8	72	88	16%	56	88	32%	65	85	20%	23%
F-15/9	63	91	28%	NA	NA	NA	74	91	17%	23%
F-15/7	61	92	31%	70	89	19%	76	93	17%	22%
F-15/4	74	86	12%	64	83	20%	56	90	34%	22%
F-16/2	62	85	23%	75	93	18%	68	86	18%	20%
F-16/5	63	88	25%	72	85	13%	68	88	20%	19%
F-15/6	70	85	15%	73	82	9%	70	89	19%	14%
F-15/3	74	89	16%	78	91	13%	93	90	-3%	9%
Mean	62	87	25%	64	90	26%	66	90	24%	25%
Median	63	88	25%	62	93	31%	65	90	25%	24%

Training Effectiveness

Aggregate data regarding knowledge gains from a sample of 183 students at small group tryouts shows overall knowledge gains (from the first three modules) average 25% (see Table 1). The data from these three modules reflect testing of the background knowledge required for troubleshooting. This amount of increase is not unexpected and is of no particular significance, based on the author's experience, other than showing that considerable learning took place.

Note that the pretest scores by module average 62%, 64%, and 66% respectively, else the increase in knowledge scores may have been higher. This can be attributed primarily to two factors. First, the students had a fairly high entry knowledge, since they have already graduated from formal technical training and had flightline experience. Secondly, all tests are "open-book" with no time limits, so students had the opportunity to research their answers. This ready access to technical data is *intentional* since the students are *required* to use the technical data in actual flightline performance of all job tasks.

The increases in performance as measured by the simulations used for pretests and progress checks in the Troubleshooting modules appear to be quite significant. Only 14% of the students could solve the problems in the pretest while 93% solved them in the progress checks, an increase of 79% (see Table 2). The low pretest scores appear especially significant given that the students are supplied and encouraged to use technical data (whicH often includes very specific fault-isolation guides) during the *entire* course.

A general finding is that while students may know "facts and figures" about a system, they still do not truly understand how it functions -- especially when a component of the system has failed. While they may

Table 2. MCTP Performance Gains on Troubleshooting Simulations (Data From Small Group Tryouts)

Course	Pretest Attempts	Pretest Passed	Pretest %	Progress Check Attempts	Progress Check Passed	Progress Check %	Difference
F-16/1	7	1	14%	7	6	86%	71%
F-15/1	9	1	11%	8	8	100%	89%
F-16/2	8	1	13%	7	6	86%	73%
F-15/2	7	0	0%	7	6	86%	86%
F-16/3	NA	NA	NA	NA	NA	NA	NA
F-15/3	8	1	13%	7	5	71%	59%
F-16/4	9	0	0%	9	9	100%	100%
F-15/4	8	2	25%	6	6	100%	75%
F-16/5	10	2	20%	9	9	100%	80%
F-15/5	11	4	36%	9	7	78%	41%
F-16/6	12	1	8%	11	11	100%	92%
F-15/6	11	0	0%	11	11	100%	100%
F-16/7	12	8	67%	7	6	86%	19%
F-15/7	7	0	0%	7	6	86%	86%
F-16/8	12	1	8%	12	11	92%	83%
F-15/8	9	1	11%	9	9	100%	89%
F-16/9	8	2	25%	7	6	86%	61%
F-15/9	8	1	13%	7	7	100%	100%
F-16/10	8	0	0%	8	8	100%	100%
F-15/10	10	0	0%	10	10	100%	100%
F-16/11	9	0	0%	9	9	100%	100%
Totals	183	26	14%	167	156	93%	79%

Table 3. F-16 Pretest to Progress Check Score Increases Correlated by Work Unit Code -- Partial List

Course	WUC	Descriptor	Pretest (Average)	Progress Check Avg.	Difference
16/3	14BCO	Integrated Servoactuator, Flaperon	29%	100%	71%
16/2	13AAD	Handle Assembly, Landing Gear, Pilot's	38%	100%	62%
16/2	14A00	Panel Assembly, Digital Flight Control	43%	100%	57%
16/4	74AWO	Cable, Transmit Drive	44%	100%	56%
16/4	74CEO	General Avionics Computer	28%	83%	55%
16/4	74KAO	Multifunction Display	47%	100%	53%
16/7	27GMO	Hydraulic System	42%	94%	52%
16/8	75B00	External Stores	32%	83%	51%
16/2	14DAO	Leading Edge Flaps	50%	100%	50%
16/3	14BAO	Integrated Servoactuator, Rudder	50%	100%	50%
16/3	14BBO	Integrated Servoactuator, Horizontal Tail	50%	100%	50%
16/3	14A00	Panel Assembly, Digital Flight Controls	42%	92%	50%

Table 4. Pretest to Progress Check Score Increases Correlated by Task (From STS or JQS) -- Partial List

Course	Core Task	Pretest (Average)	Progress Check Avg.	Difference
16/2	Troubleshoot Aircraft Wiring	39%	100%	61%
16/2	Use Data Transfer Terminal Switch	39%	100%	61%
16/5	Rig Engine Power Control System	40%	100%	60%
16/1	Use Jet Fuel Starter (JFS) Analyzer	34%	90%	56%
15/2	Perform Operational Checkout on Manual Flight Controls	45%	100%	55%
16/3	Use Portable Hydraulic Test Stand	42%	95%	53%
16/4	Troubleshoot RF Wiring	48%	95%	47%
16/3	Trace Signal/Data Flow in DFLCS	53%	100%	47%
15/[illegible]	Trace Wiring/System/Interface on Manual Flight Controls	47%	94%	47%
[illegible]	Troubleshoot External Fuel Tanks System	53%	100%	47%
[illegible]	[illegible] Faulty Antennas	41%	87%	46%
[illegible]	[illegible] Hydraulic Test Stand - Support Equipment	50%	95%	45%

know how to perform a given troubleshooting procedure, they do not understand when and why to perform it in the context of actual problems -- they cannot *efficiently* select which procedure to use in specific circumstances. These findings are substantiated by the high pretest scores on the previous modules (especially the Diagnostics module, which covers diagnostic tests and test equipment), yet poor performance on the Troubleshooting pretests. Note that the modules (including pretests and progress checks) are presented in sequence, so the students have a firm knowledge foundation prior to taking the Troubleshooting pretest simulations. Yet, only 14% of the students could pass these problem-solving pretests.

Cost Effectiveness

As discussed earlier, the small sample size from small group tryouts does not support conclusive analysis of data on a specific course. However, this preliminary data is quite encouraging. When analyzed by Work Unit Code (i.e. a maintenance action related to a specific component, assembly, or subsystem), twelve F-16 components show a troubleshooting knowledge increase of 50% or more (see Table 3). Dozens show more than a 30% increase. Similarly, nine F-16 tasks show a 45% or more increase (see Table 4). Should these preliminary indications bear out, savings related to just one F-16 maintenance task (e.g. Isolate Faulty Antenna for the APG-68 Radar, with a 46% increase as in Table 4) or reductions in supplies of just one F-16 component (e.g. General Avionics Computer, with a 55% increase as in Table 3) could feasibly pay for this entire courseware development effort.

Subjective Feedback from Critiques

Analysis of the aggregate data from the standardized, automated, Air Force-provided critiques is quite positive overall (see Table 5). It averages 3.97 on a 5-point Likert scale (1 = low, 3 = average, 5 = high). There was a wide range in critique input when analyzed by question and by course, as shown by the High and Low scores. When all questions were summarized by course, the overall evaluation of each course ranged from a low of 3.36 for course F-15/8 to a high of 4.43 for course F-16/7, with a mean of 3.97 and a median of 3.95. The lowest evaluation on 10 of 16 questions and the lowest overall evaluation came from course F-15/8 -- which had the highest knowledge gains recorded.

Note the overall scores of 3.76 on Question 1 and 3.94 on Question 7, yet overall scores of 4.11 on Question 15 and 4.19 on Question 19. Though the students appeared to think that the courses did not prepare them for actual job performance as well as they could have (though 3.76 on a 5-point scale is still quite positive), they felt that they were relatively better prepared for the progress checks (3.94). Further, the courses were rated both very beneficial personally (4.11) and very valuable for further review (4.19). Perhaps this disparity between perceived value and the lower score on preparation for job performance can be attributed to a desire for more practice on more problems.

Table 5. Critique Results From Small-Group Tryouts (Air Force-provided questions, 5-point Likert scale)

Question:	High	Low	Avg.
1. How well did the course prepare you for job tasks taught in the course?	4.54	3.08•	3.76
2. How well did the course hold your attention?	4.56	3.33•	4.01
3. How well is the course divided into "bite-sized" segments of instruction?	4.80	3.17•	3.99
4. How easy is it to exit and return to the course?	4.77	3.63	4.13
5. How clear were the directions for using the course materials?	4.67	3.42•	4.13
6. How well did the pretest results allow you to bypass parts of the course you already knew and did not need?	4.33	3.00•	3.79
7. How well did the course prepare you for the progress check?	4.62	3.25•	3.94
8. How well did the progress check test what you learned in the course?	4.62	3.25•	4.00
9. How clear were the test questions?	4.15	3.25•	3.70
10. How adequate (free from noise and interruptions) is the interactive courseware workstation?	4.33	3.09	3.93
11. How well did the course encourage use of appropriate T.O.s or other technical data?	4.54	2.71	3.86
12. How easy was it to get a copy of the technical data needed to take the course?	4.62	3.25•	4.16
13. How available was the course for use during your duty hours?	4.60	3.13	3.84
14. How well did the course hardware and software operate?	4.70	3.25•	4.12
15. How beneficial was the course to you personally?	4.85	3.40	4.11
16. How helpful do you think it might be to have this course available for your review at some future date?	4.92	3.53	4.19
		TOT	3.97

• From course 15/8, whose knowledge increase was the highest recorded (41%).

Best Available Copy

CONCLUSIONS AND RECOMMENDATIONS

Methodology Issues. The MCTP effort validates the widespread utility of the PARI process, especially for less experienced interviewers. The author can recommend the PARI process without reservation for those instructional simulations where the learners' actions are the key input to the outcome -- as opposed to a process simulation, for example. [See Alessi and Trollip (1991) for an excellent description of the types and uses of instructional simulations.]

The validation of tests and the analysis of results prior to designing and developing the instruction helped significantly in focusing the training on the most needed areas. Further, it also provided considerable data that was incorporated directly in the course design. While most instructional design models call for developing the tests prior to designing the instruction, none (to the author's knowledge) describe the benefits of actually administering it and analyzing the results. This practice should be considered for more widespread use, since it does not take that much additional effort yet it yields significantly valuable data.

With performance improvements more than three times the size of the knowledge gains, the data would suggest that the courses are quite effective in teaching students *how to apply* their knowledge to complex troubleshooting problems. The data would also suggest that while essential for measuring troubleshooting skills, simulations are not the *only means* of teaching troubleshooting. When combined with carefully designed tutorials, a more cost-effective mix of instructional strategies may be possible. Finally, the data would support the proposition that mental models *can effectively be taught directly* rather than through inference or self-discovery. More research and evaluation of this issue is definitely warranted, given the development cost of simulations as compared to tutorials. A later enhancement of this simulation development methodology used for a commercial client (troubleshooting electrical problems on diesel/electric locomotives) was to add a "Rationale" section to the summaries presented after each simulation. This section grouped the actual steps performed in a troubleshooting sequence into a higher level of "key tests or checks," explaining the experts' rationale for performing them. (Often, it requires several distinct steps to perform one complete test in order to split the problem space.) This enhancement appears to reduce the amount of practice needed for the novice to understand and start modeling the experts' performance, yet this was not a controlled study so any conclusions would be premature.

The data would also suggest that psychological (i.e. cognitive) fidelity is the key component of simulation fidelity for complex maintenance tasks rather than physical or functional device fidelity. Note that these courses were developed on an 80286 microcomputer, 640 kB RAM, dual floppy drives (no hard disk), CGA graphics, and an interactive videodisc player. Physical realism was limited to visuals of actual equipment, and these were used primarily to depict the results (and not the possible actions) as mentioned previously. There was no true physical fidelity. Functional device fidelity was also quite limited. Care was taken to depict accurate results for any of the action choices provided. However, these action choices were limited to the actions performed by either novices or experts during the cognitive task analysis -- likely actions, not all possible actions, were simulated. This resulted in a *much simpler simulation model* than exists in either computer-driven part-task trainers or intelligent tutors that are used to present the same (or similar) content. This impacts the design, development, delivery, and maintenance costs of the simulations. While further research is definitely needed, this data would suggest that significant increases in the cost-effectiveness throughout the life cycle of maintenance troubleshooting simulations are possible by using cognitive task analysis techniques to build high psychological fidelity.

Research Issues. The author would echo the opinions of Lane and Alluisi that discussions of fidelity are confounded by the terminology: "...unless we add a great many additional modifiers, the term fidelity is so general as to be almost meaningless in discussing simulations" (1992, p5), and further that paying the high price of high fidelity does not ensure training effectiveness: "...all the fidelity you can afford may be *too much* (their emphasis) for optimum training (1992, p10)." The author would also contend that the issue of fidelity is further compounded by the difference in maintenance versus operator tasks. *Most published research seems to apply to operator training, yet many seem to apply it directly to maintenance.* Most maintenance tasks do not seem to require the precise psychomotor reactions, the constant monitoring of a myriad of visual, auditory, or tactile cues, the real-time event simulation, nor the potentially unlimited emergency conditions that are common in operator tasks.

The author would encourage more specific research on instructional simulations to address the following questions:

1. "How much simulation is enough" to reach the *most cost-effective mix* of instructional strategies? Under what conditions and for what audience? While simulation appears essential for some things, it is unquestionably the most expensive instructional strategy being widely used in interactive courseware. When must it be used and what alternative strategies are both efficient and cost-effective? How much practice is optimum for cost-effectiveness, both in terms of actual learning and to provide sufficient student confidence to facilitate transfer to the job environment?

2. How much fidelity (and of what type) is needed for the *most cost-effective development of simulations?* For what subject matter and audience? Just as in "hardware simulators," the fidelity of simulations in interactive courseware is the primary determinant of cost. Again, how much is enough?

3. What are the quantifiable differences (if any) between the requirements for *operations versus maintenance simulations?* How do they impact fidelity requirements? As discussed above, maintenance and operator tasks seem to differ substantially in some of the key areas related to the fidelity of simulations. For example, maintenance troubleshooting simulations seem to have more in common with medical diagnostic simulations than with flight simulations.

4. What are the quantifiable differences (if any) in fidelity requirements between maintenance *initial skills training and advanced troubleshooting training...* at least in the key areas of physical, psychological, and device/functional fidelity? It appears that high physical fidelity and low functional fidelity is needed for initial skills training, while high psychological fidelity is necessary for advanced training. Building either unneeded functional fidelity into initial skills training systems or physical fidelity into advanced training systems significantly increases both complexity and cost. How much is truly needed?

Management Issues. In conclusion, the author would strongly recommend that the data from implementation of this courseware be promptly gathered, analyzed, and compared with CAMS maintenance data, then widely disseminated. In addition to more insight into the instructional design and training effectiveness issues above, the cost-effectiveness data could be of significant benefit for other potential courseware programs, as follows:

1. Reductions in Maintenance Manhours or Spare Parts. These two areas were originally targeted as the potential payback for this training investment in the PIF package (as discussed earlier). Most other investments in interactive courseware use reductions in training time, instructor salaries, training facilities, or travel costs as the potential offsets. Instead, the MCTP attempts to tie the return on training investment back to operations and maintenance costs, a much larger potential return on investment (potentially more than a 100:1 payback ratio) and an area directly related to mission capability. If successful, this attempt could provide both a significant, large-scale precedent and a proven methodology in how to do so -- allowing others to use similar training investment strategies.

2. Reductions in Depot-Level Repairs. Aircraft wing commanders must now pay for depot-level repair of ALL components processed by a maintenance depot, whether they are defective or not. This was not the case when the MCTP was originally proposed and funded. Should the data confirm that this troubleshooting training decreases those depot repair costs (as it should, since it should reduce the number of components removed erroneously), the local wing commander can have more budget flexibility and will likely be a stronger proponent of training. "Two-level maintenance" (as described earlier) will likely increase the cost of depot-level repairs (and the accompanying transportation costs) otherwise, since there will be no local "check" to prevent components being sent to the depots unnecessarily.

One of the persistent challenges of training efforts has been to show a direct relationship of training results to either mission capability or local management initiatives. The results of training and the accompanying impact on performance have often been either intangible or difficult to measure. The MCTP's automated analysis programs and the data that they generate could demonstrate a methodology for providing the local commander (and higher levels of management) quantifiable results (in both time and money savings) in direct relationship to their emphasis on and support of training.

Based on the preliminary results of this MCTP project and the author's experience, the answers to some of

these issues may not be as obvious as they seem. Meanwhile, those of us frequently tasked to "transition research into reality" sure need to know...and to have the data to support and justify our program management actions or instructional design decisions.

REFERENCES

Alessi S.M. and Trollip, S.R. (1991). Computer-based instruction: Methods and development. Englewood Cliffs NJ: Prentice-Hall.

Fletcher, J.D. (1990). Effectiveness and cost of interactive videodisc instruction in defense training and education. IDA paper P-2372, prepared for Department of Defense. Alexandria VA: Institute for Defense Analyses.

Hall, E.M., Gott, S.P. and Pokorny, R.A.. (1990). A procedural guide to cognitive task analysis: The PARI methodology. Draft AFHRL Technical Paper 90-XX, Brooks AFB TX: Air Force Human Resources Laboratory, Manpower and Personnel Division.

Gott, S.P. (1989). Apprenticeship instruction for real-world tasks: The coordination of procedures, mental models, and strategies. In E.Z. Rothkoph (Ed), Review of Research in Education, Vol 15, 97-169. Washington DC: American Educational Research Association.

Hannafin, M.J. and Rieber, L.P. (1990). Psychological foundations of instructional design for emerging computer-based instructional technologies: Part I. Educational Technology Research & Development, 37(2), 91-114.

Lane N.E. and Alluisi, E.A. (1992). Fidelity and validity in distributed interactive simulation: Questions and answers. IDA Document D-1066, prepared for Department of Defense. Alexandria VA: Institute for Defense Analyses.

Lesgold, A.M., Lajoie, S.P., Bunzo, M., & Eggan, G.M. (1988). Sherlock: A coached environment for an electronics troubleshooting job. Paper presented at the J.S. Macdonald Foundation Conference on CAI and ITSs, Carnegie-Mellon University, Pittsburgh, PA.

Thomas, T.K. (1987). The use of interactive videodisc training systems for tactical aircraft maintenance training. Proceedings of the Fifth Annual Technology and Innovations in Training and Education (TITE) Conference. [illegible] US. U.S. Air Force.

Wilson, J.R. and Rutherford, A. (1989). Mental models: Theory and application in human factors. Human Factors, 31(6), 617-634.

MULTIMEDIA INFORMATION RETRIEVAL - REVOLUTIONARY RESULT OF TECHNOLOGY FUSION

Scott D. Royse
Denise C. Varner, Ph.D.
Bruce C. Mather, Ph.D.
Southwest Research Institute
San Antonio, Texas

Abstract

In his 1992 article in Harvard Business Review, Fumio Kodama defined technology fusion as the "nonlinear, complementary and cooperative blending of incremental technical improvements from several previously separate fields of technology to create products that revolutionize markets."[1] This paper describes the design and application of a multimedia information storage and retrieval system that is the blending of digital multimedia, database management, and communications technologies. The resulting system has demonstrated the potential for dramatically changing the ways in which computer systems are applied to accomplishing work. As the multimedia capabilities of PCs become as common as math coprocessors are now, this new method of information management will blur the lines of distinction between training and work, and will add new dimensions of meaning to the concepts of "computer-based training" and "embedded training."

The Visual Information System (VIS) is a multimedia data management system with an intuitive, visually oriented user interface. Each node in the data structure may have multiple information elements that may be photographic, computer graphic, video, animation, audio, text and numeric. In addition, user-generated notes and tutorial programs may be attached to any node in the database, and initiated at the user's request. Database navigation may be accomplished either by linear traversal of the data structure, by directed search according to specified criteria, or by hyperlinks to other data records.

This paper will demonstrate applications of the VIS concept to aircraft systems (MH-53J Pavelow Helicopter) and electrical cable and connector repair, and will describe the system (hardware and software components) and how it works. The paper will conclude with a discussion of other potential applications that focus on why this startling new capability is important.

ABOUT THE AUTHORS

Scott D. Royse is manager of the Training Systems Development Section at Southwest Research Institute. He has over 12 years of experience in systems engineering projects for automotive and missile electronics, and has developed simulator training systems for military and industrial clients. He has managed many training device development efforts including the Combined Arms Staff Trainer for the USMC. He has a BSEE from Texas A&M University.

Denise C. Varner is a Principal Scientist in the Advanced Training Concepts Section at SwRI. She specializes in research on human-computer interfaces including visual databases, speech recognition, and virtual environments. She trained as an experimental psychologist, and holds a BA from Florida State University and an MA and PhD from the University of Pennsylvania. Dr. Varner is the author of more than 30 papers and publications.

Bruce C. Mather is a Senior Research Engineer specializing in computer architecture, computer systems design (hardware and software), and multimedia. He holds a BS, MS, and PhD in Electrical Engineering from the University of Illinois at Champaign-Urbana. He has worked at SwRI for seven years in diverse fields including robotic systems, image processing, digital signal processing, biomedical instrumentation, artificial intelligence, and neural networks. His current research areas include virtual reality and multimedia database interfaces.

MULTIMEDIA INFORMATION RETRIEVAL - REVOLUTIONARY RESULT OF TECHNOLOGY FUSION

Scott D. Royse
Denise C. Varner, Ph.D.
Bruce C. Mather, Ph.D.
Southwest Research Institute
San Antonio, Texas

INTRODUCTION AND OVERVIEW

The year is 1983. Charlie Chaplin is on television selling the IBM PC and AT. The PC Jr. emerges to address the home market, and Lotus 1-2-3 and Wordstar are the kings of the desktop. Mailmerge is an exciting capability and a 20-Mbyte hard disk is a big deal. . . .

The year is 1993. It is the age of digital multimedia, Quicktime, and Video For Windows. Electronic mail and networking are commonplace, as are cellular telephones, personal digital assistants, and telephones in airplanes. The arrival of Pentium is looming, and 486s can be purchased at WalMart for $995. . . .

The year is 2003. Is our imagination robust enough to visualize the exciting results of another decade of development in computer and communications technologies? It seems clear that the applications supported by inexpensive desktop systems will continue to revolutionize the workplace as the engine of the information age.

This paper will describe a multimedia information management system that addresses information processing tasks of proficient users, as well as the training tasks associated with new users, or as may be associated with new procedures. The blending of improvements in digital multimedia and database management technologies with improvements and standardizations in computer systems and operating environments results in applications with capabilities that will change dramatically the concepts of Computer-Based Training (CBT) and Embedded Training, and will alter fundamentally the way computers are used to accomplish work. Maturation of the concepts illustrated here in combination with a continuing integration of computer and communications technologies could lead to information systems used on-the-job both to accomplish work and to manage continuous improvement training programs that account for individual learning styles.

MULTIMEDIA INFORMATION RETRIEVAL—HOW IT IS USED

Application Example—Aircraft Systems Database

One example of this type of multimedia information management system is a database system, illustrated in Figure 1, established for use by a USAF Special Operations Forces (SOF) aircraft support group. The Visual Information System (VIS) provides a centralized, on-line reference source for accessing system and component information on the MH-53J Pavelow helicopter, and was established to improve the productivity of experienced personnel in the group and to decrease the time required for new personnel to become proficient in their duties. To accomplish both objectives, the system must incorporate features that address the information management tasks of proficient users as well as the training needs of newcomers. The system was designed for use by the following types of users, each with their own unique job task duties, aircraft system areas of concern, and methods of accessing and referring to aircraft data.

- **Program managers'** principal job tasks are to set and communicate priorities, coordinate activities and monitor project performance. They must have a high-level understanding of the aircraft configuration and the interrelationships between aircraft systems.

- **Item managers** provide organizational accountability for supply, provisioning, installation, and maintenance of specific components or systems. They must have an in-depth knowledge of stock number history and parts usage and availability of the components and systems for which they are responsible.

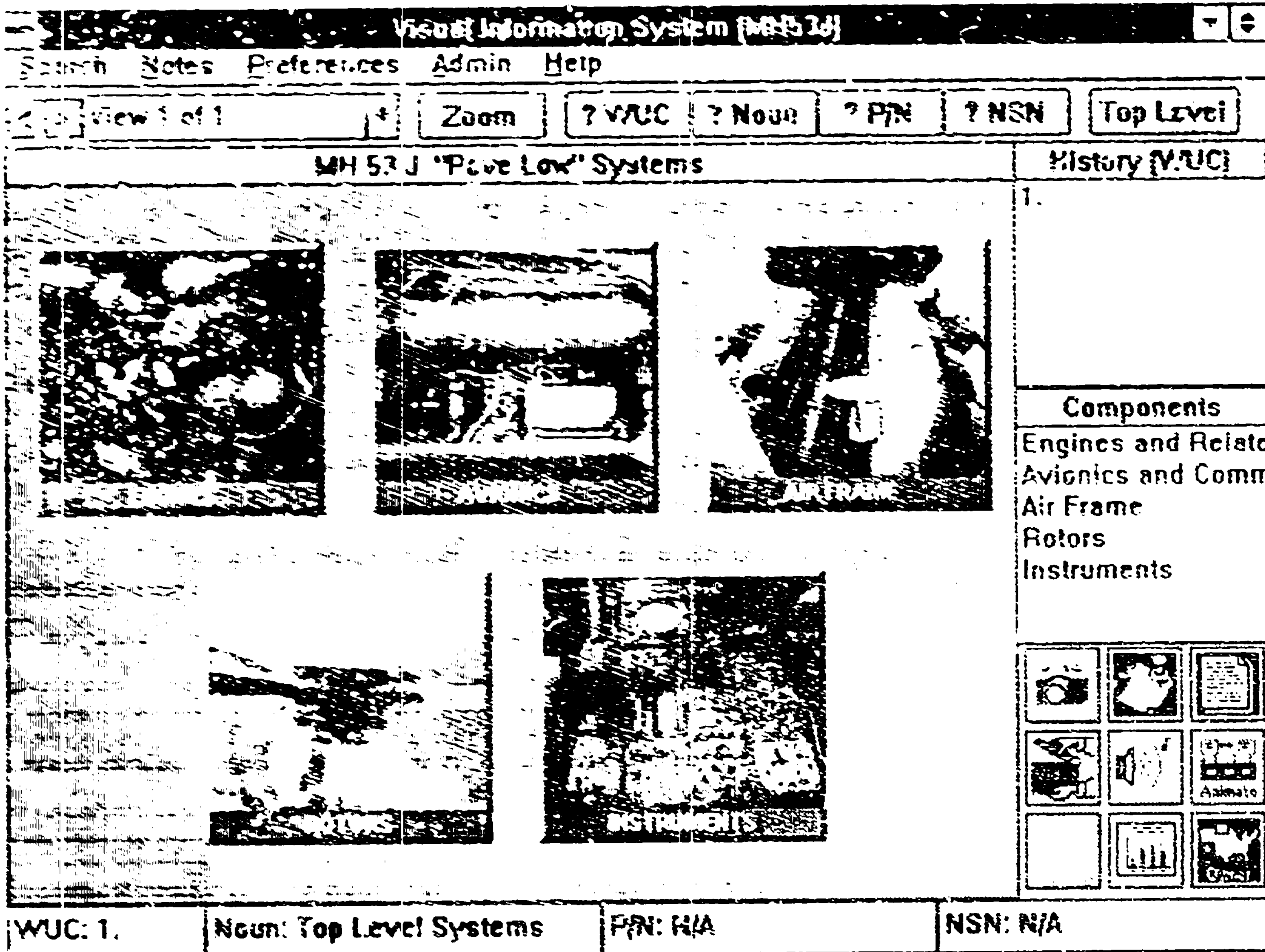


Figure 1. Top Level VIS Screen

- **Engineers** are responsible for evaluating system and component failure modes and assessing the impact of design modifications. They must have an in-depth knowledge of aircraft operations and system and component functions.

- **Equipment specialists** are responsible for functional and physical interfaces between subsystems and for the change activity associated with specific aircraft systems. They must analyze the statistical history of aircraft components in order to project current and future needs.

- **Contract specialists** must have an overall understanding of the aircraft systems to evaluate Statement of Work requirements. They must also visualize the items of a purchase request package.

The job tasks and information requirements of these individual user groups break down into the three basic aircraft configuration areas of interest shown in Table 1.

The VIS addresses these information needs by providing access to multimedia aircraft information that is stored in a logical hierarchical structure (the classic inverted tree structure) that allows a user to navigate freely through the aircraft data or to query on a specific system component. Information is collected and stored for each aircraft system from a variety of sources, and the system allows each individual user to associate annotations with any information node of the database. In order to address the special needs of newcomers to the organization, tutorials, guides and procedural job aids may also be attached to any information node of the system, and launched at the discretion of the individual user.

Application Example—Intelligent Computer-Aided Cable Repair System

A second example of the application of multimedia information retrieval is the Intelligent Computer-Aided Cable Repair System (ICACRS), illustrated in Figure 2. The objective of the ICACRS is to decrease the time spent by U.S. Air Force

Table 1. VIS User Information Requirements'

Job-Related Area of Interest	VIS Media Requirement	Media Requirement Rationale
System Operation and Function	• Motion video with audio • Animated diagrams with correlated audio • Engineering drawings • Text	• See actual system in operation • Show/describe component function • Show component design • Present supporting information
Component Change History and Integration Affectivity	• Annotated photo • Static diagrams with correlated audio • Text • Statistical graphs	• See actual system/component • Show/describe system interfaces • Present supporting information • Show visual change history
Configuration Ownership	• Annotated photo • Graphic charts • Text	• See actual system/component • Show owner/aircraft relationships • Present supporting information

maintenance personnel in identifying and repairing aircraft cable and connector components by providing an integrated resource for identification, technical documentation, and repair procedures on cables and connector components used in aircraft and test equipment. Multimedia information presentation capabilities provide the technician with connector and cable identification aids, technical specifications and diagrams, and step-by-step tutorials on the disassembly, repair, assembly, and test of connector/cable components. The utility of the ICACRS can be fully appreciated when one realizes that there are over 750,000 unique part numbers just within the Mil-Spec families of connector items. Commercial grade connector families add several hundred thousand more part numbers to the total that must be represented in the database.

The system will be used by personnel with a wide range of maintenance experience, from expert technicians through entry-level maintenance personnel. The system is designed to accommodate this wide spectrum of users by providing the means by which experienced users may obtain data in a minimum number of steps, while an inexperienced or infrequent user may use additional features to obtain more detailed information. Tutorials can be used as refresher training for those tasks that are not performed frequently, and are valuable for the inexperienced and infrequent user. The tutorials also provide a mechanism for formalizing on-the-job training (OJT). For both Reservists and entry-level technicians, the system provides an electronic user's manual as well as on-line and context-sensitive help to assist them with their tasks. The ICACRS concept of applying multimedia information retrieval techniques to establish such an integrated resource is particularly appropriate for the anticipated migration in the Air Force to a two-level maintenance concept.

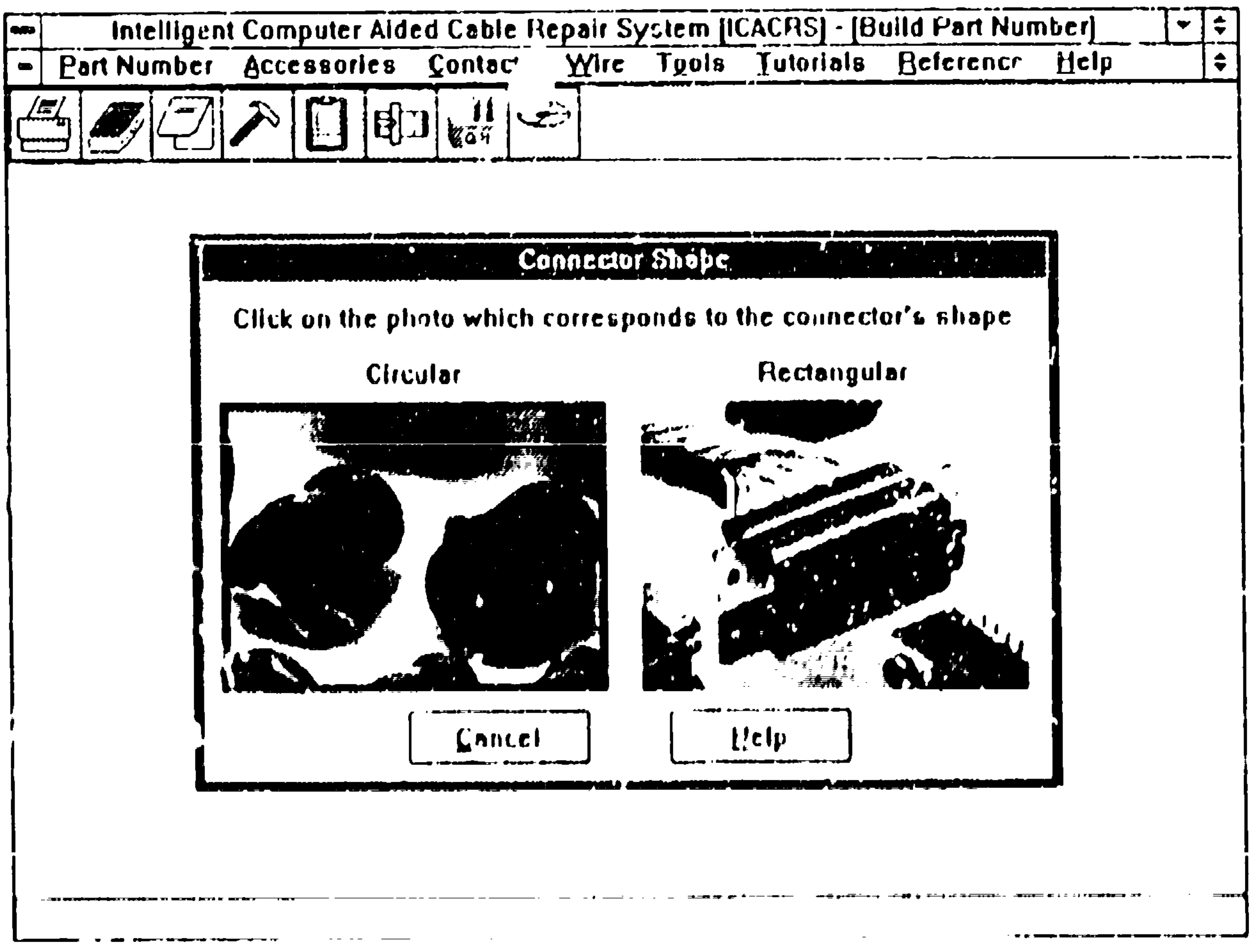


Figure 2. Intelligent Computer Aided Cable Repair System

MULTIMEDIA INFORMATION RETRIEVAL—HOW IT WORKS

The VIS and ICACRS are powerful applications of multimedia database technology because they successfully blend the intuitive user interface standards of the Windows operating system with the information-rich presentation capabilities of multimedia data. Standard Windows interface characteristics simplify user interaction with the system to the point that database traversal and information retrieval become obvious. The resulting systems provide quick and efficient access to information and the ability to partition learning tasks into small, manageable, context-sensitive modules. The systems result from a blending of database, multimedia and computer technologies as described in the following paragraphs.

Database Structure, Navigation and Query

Using the Visual Information System as an example, the database is organized into an inverted tree structure of information nodes, as shown in Figure 3. This is no different from the classic database structure wherein each node of the tree has a single parent node, and may have multiple daughter nodes. However, instead of being confined to alphanumeric records, each information node may have an unlimited number of data records, with each record consisting of a file of a specific multimedia data type. Thus, each information node may have a multitude of data elements that are text, images, video, audio, or any of the other multimedia data types presented by the system. Furthermore, information at a given node can be directly linked to another node by visual reference, providing powerful capabilities for database navigation.

Database navigation mechanisms that allow users to rapidly traverse to a specific information node from which a query may be launched can be summarized by the words traversal, hyperlink and history. Traversal implies that navigation may progress relative to present position by linear movement up and down the parent-daughter links of the hierarchy. This may be accomplished in several ways that are both textually and visually oriented, and provides an inherent browse characteristic in the system. Hyperlink implies the ability to move from the present position in the data structure to any other point that is unrelated in terms of direct parent-daughter relationship.

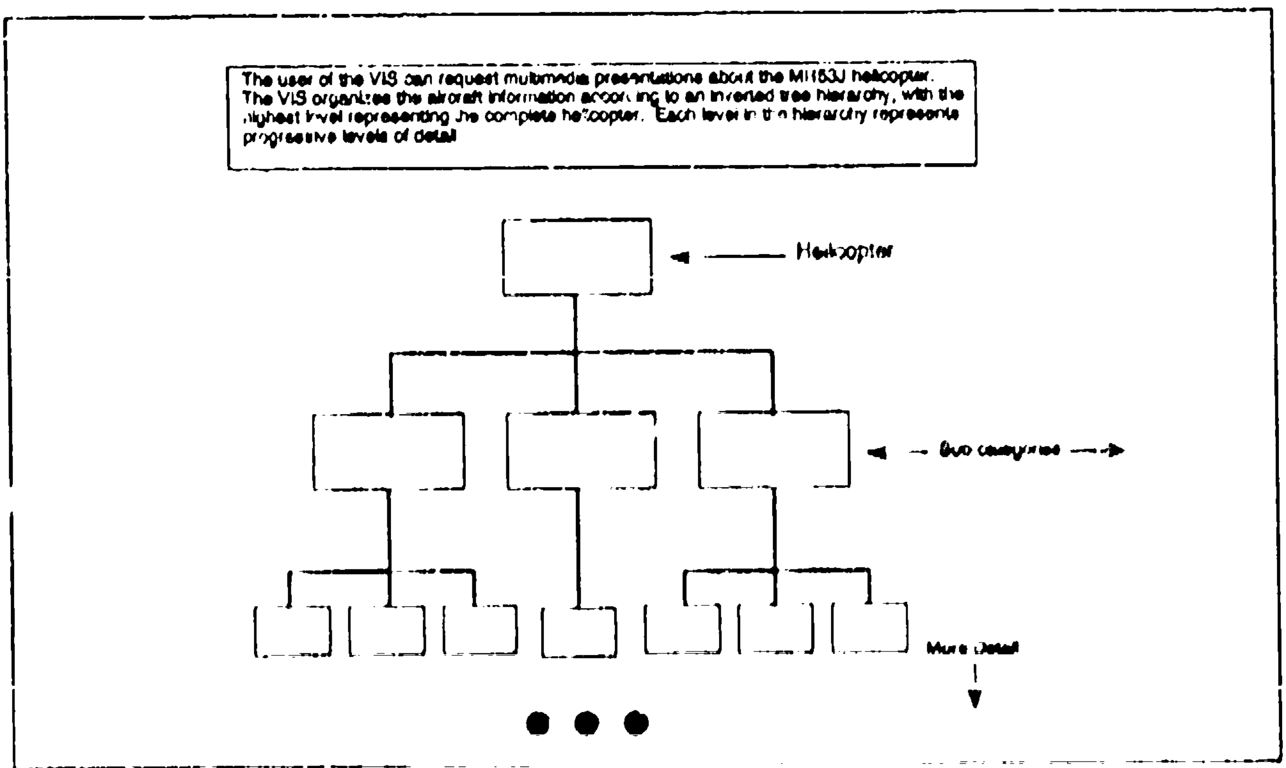


Figure 3 VIS Structure of Visual Data

Hyperlink is accomplished primarily by searches which may be directed to a specified part number, nomenclature (proper name), stock number or work unit code. Hyperlink may also be accomplished by a text search to localize into a specific area (i.e., "radar"), with the exact information node then being selected from a "short list" of candidates. Finally, hyperlink may also be accomplished by defining regions of images as "hot spots" that provide a visual hyperlink to other information nodes of the database. History implies the ability to return to any previously visited information node. This is accomplished by keeping a history log in a window of the screen, and permitting direct return to an information node by double-clicking on its designation in the history window.

Database query is accomplished directly from the desired information node, and is illustrated in Figure 4. The fundamental data elements associated with the node, such as the name, part number, stock number, work unit code and primary image view, are displayed immediately on the screen.

Additional image views of the subject of the node can be accessed immediately using the image bar. The existence of additional information elements is indicated by the appearance of various Multimedia Information Retrieval buttons. These buttons appear at a given information node only when information of the type indicated exists. A mouse click on the appropriate button will cause the presentation of information as indicated in Table 2.

Multimedia Data Types

The defining feature of the VIS and ICACPS is the ability to store, retrieve, and display data of many types, including:

- Text (ASCII)
- Digital Audio (.wav and compressed)
- Digital Video (.avs, .avi, .qtw--hardware and software)
- Animation (.flc)
- Tutorials (Authorware, Toolbook)
- Graphics (.bmp, .wpg, .pcx, etc)
- Photos (.bmp, jpeg compression, others)

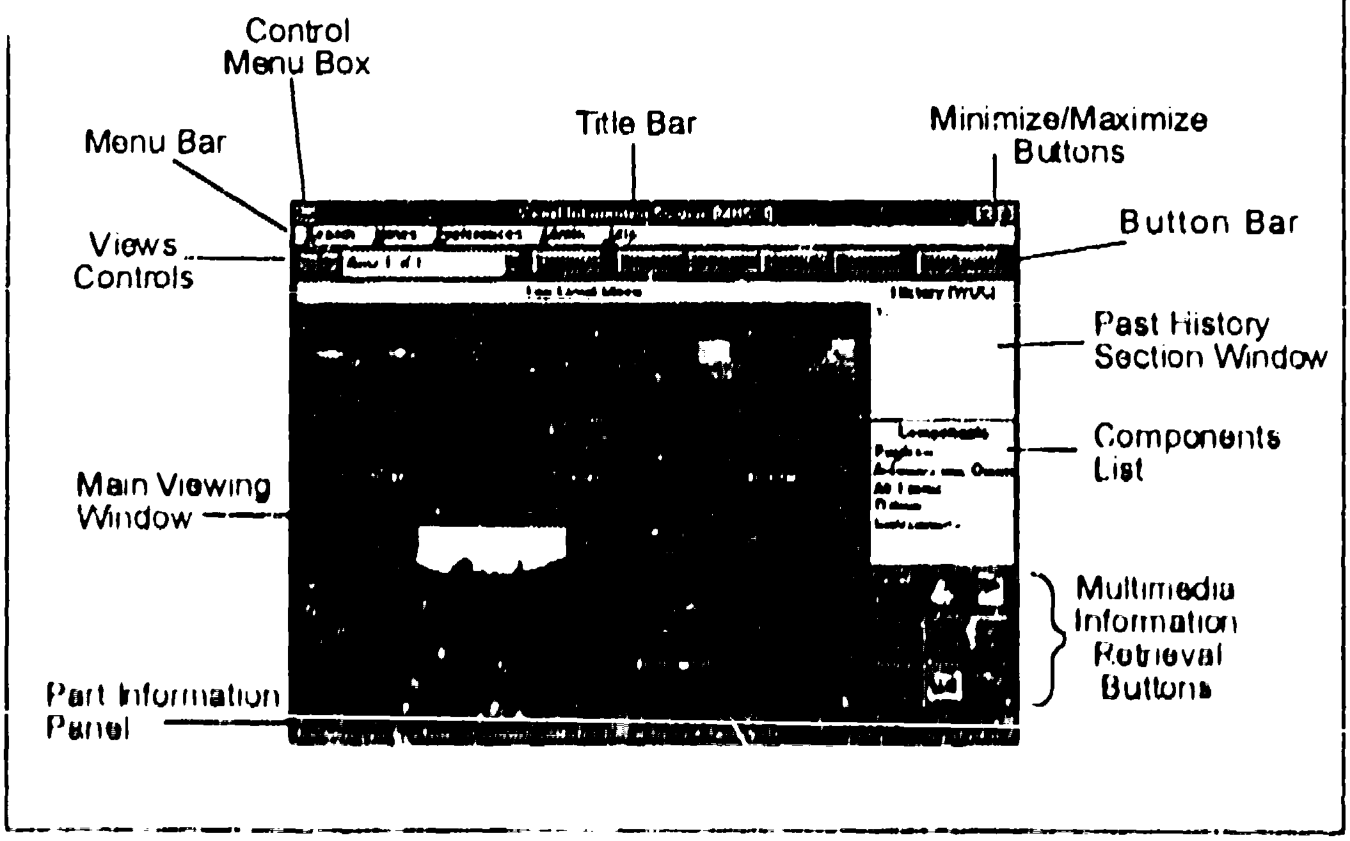


Figure 4 Database Query

Table 2. Multimedia Information Retrieval Buttons

BUTTON NAME	ICON	LOCATION	FUNCTION
Media Info	Camera 	upper left corner	gives info about an image in the main viewing window
Notes	"Stick-on" note 	top row center	indicates a note is associated with this node
Text	Document 	upper right corner	indicates a text file is associated with this node
Video	Clapboard labeled "video" 	center row left	indicates that motion video is associated with this node
Audio	Loudspeaker (an "X" appears when the icon is clicked) 	center row center	indicates that a sound file is associated with this node
Animation	Flowchart labeled "animation" 	center row right	appears whenever an animated file is associated with a node
Tutorial	Mortarboard (cap and tassel) 	bottom row left	means that there is a tutorial associated with the current node
Change History	Bar graph 	bottom row center	gives data about the change history of a part
Ownership history	Map labeled "owner" 	bottom row right	shows a map of the U.S. with the year each ALC was responsible for the part

Each data type requires an editor or other interface and, in most cases, driver software. Simplified management of so many different data types and the ability to easily add new data types to the list are made possible only by strict adherence to the standards and conventions of the Windows operating system. For example, the VIS contains two types of video files – Video for Windows (.avi) and DVI® files (.avs). These video files are displayed to the user through a common interface but have different drivers and hardware requirements. The choice of file type may be made by the system administrator and should be based on frame rate and image quality requirements of the application. DVI files (now known as "Indeo") are played through specialty hardware (Intel's Action Media II board) that can produce 30 frames-per-second video at one-quarter VGA resolution. Video for Windows has a smaller native screen resolution and variable frame rate but is software only. Both video types, however, are played on a screen with a television set visual metaphor, as shown in Figure 5, with controls that are common to VCRs. From the user's point of view, the file type is not known and is irrelevant. QuickTime for Windows video files (.qtw files) or other standard video file formats can be added easily so long as Windows can accept the driver for the file type.

For text, a simple editor has been written so that the data administrator, with proper privileges, can easily make edits but an ordinary user can only make limited changes (by means of the "Notes" utility on the Multimedia Information Retrieval button pad). For most other data types, a custom interface has been designed but with reliance on third party software drivers such as Animator Pro, Authorware Professional, or PhotoStyler. In this way, users are provided with access to visual information quickly and easily in a standard format, but system administrators may use the Windows compatible editing or composing tool of their choice. Adding new data types requires only installing a new Windows compatible driver.

Hardware and Software Components

Multimedia information applications such as the ICACHS and the VIS can be run on 386 machines with 20 mHz clock rates and 4 mB of RAM. However, the applications function best in a 486-33 environment with 8-12 megs of RAM. System disk requirements vary with the application. For video-intensive applications or for applications involving thousands of images, high storage capacity hard disks (500 mB or more) are recommended. Video capture and playback accelerator boards are also recommended for applications requiring users to view a lot of video.

As for future possibilities, recall that in 1983 20 Mbytes of disk was an astounding amount of storage space, and AT stood for advanced technology. With disk storage densities continuing to increase and the next generation of processors already becoming available, it would seem that the tools to support even more sophisticated applications are rapidly becoming available. No specific recommendations are made at this point because hardware is changing rapidly. New video accelerator boards are coming into existence monthly and should be researched carefully. However, Video for Windows has emerged as a clear industry standard for multimedia PC computing and has been included for that reason. Hardware acceleration combined with Video for Windows makes quarter-screen, 30 fps video possible on a 486 platform, and a Pentium platform may well provide software-only 30 fps video.

CONCLUSION – WHY MULTIMEDIA INFORMATION RETRIEVAL IS SO IMPORTANT

The multimedia applications described here are only examples of the truly revolutionary information management applications that are rapidly becoming reality, and will change the way computer and communications technologies are used to accomplish work. These applications are remarkable for several reasons. First, they reduce the information retrieval overhead on students and other users; they are intuitive and efficient. Second, they allow for true continuous improvement. In other words, when does the user stop working and begin training, and vice versa? The boundary between the two activities is so subtle it is almost nonexistent. Finally, the ability of these applications to integrate with other Windows products and tools provides maximum flexibility and personalization, with a standardization that makes information accessible by more people. Combine all of these concepts with such things as ISDN (Integrated Services Digital Network), wireless LANs (Local Area Networks), cellular communications, and

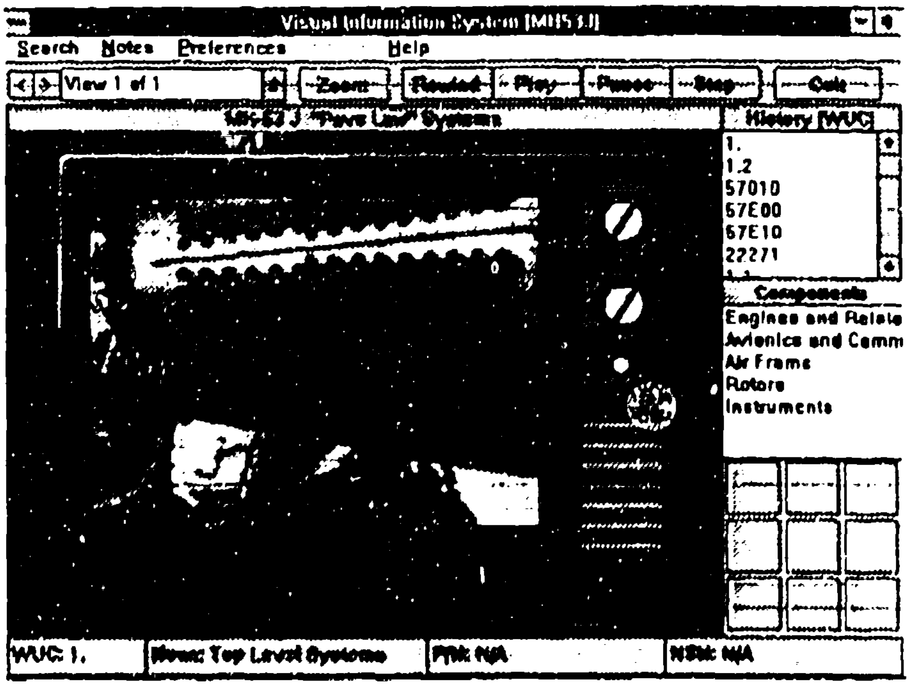


Figure 5 Video Screen

another decade of computer and communications technology development, and imagine the endless applications. . . .

- Multimedia magazines delivered daily via wireless network
- Multimedia E-mail and conferencing
- Multimedia presentations, proposals, and dissertations
- Multimedia merchandising, retailing, and advertising
- Entertainment such as interactive storytelling for children
- Technical documentation such as Interactive Electronic Technical Manuals (IETMS)
- Multimedia references including the I/ITSEC conference proceedings. At last, we will be able to prowl the exhibit floor and still experience the paper presentations!

Finally, imagine the powerful combination of this information technology with the organization concepts of Total Quality Management (TQM), focus groups, Statistical Process Control (SPC) and other developments. The potential for changing both the way computers are used to accomplish work and the concept and conduct of training is powerful.

REFERENCES

1. Kodama, F. (1992). "Technology Fusion and the New R&D," *Harvard Business Review*, July August, p. 70.

2. Southwest Research Institute (1991). "User Requirements Report," Visual Information System Project, SwRI Project 05 4373, San Antonio, TX, November, p. 6.

3. Southwest Research Institute (1993). "Visual Information System User's Manual," SwRI Project 05 4373, San Antonio, TX, May, p. 10

DESIGNING ELECTRONIC PERFORMANCE SUPPORT SYSTEMS

Susan Varnadoe
Analysis & Technology
Orlando, Florida

Ann L. Barron
University of South Florida
Tampa, Florida

Abstract

Electronic Performance Support Systems (EPSS) are designed to provide information, training, and resources to users on an "on demand" basis. This approach differs from traditional computer-based training systems in their organization, the amount of control the users maintain, and their integration with an on-the-job context.

The design of an EPSS is quite different from the design of computer based instruction. Although an overall menu structure may exist, the user generally has a great deal of freedom to move around in the system and access specific parts. In addition, hyperlinks usually exist to connect multimedia and textual resources. This paper provides guidelines and suggestions for the design and development of electronic performance support systems for maintenance and trouble shooting procedures.

DESIGNING ELECTRONIC PERFORMANCE SUPPORT SYSTEMS

Susan Varnadoe
Analysis & Technology
Orlando, Florida

Ann E. Barron
University of South Florida
Tampa, Florida

Introduction

During past years industry has witnessed a major change in corporate and industrial "desktops." The vast majority of employees now have ready access to computers, and the traditional "inboxes" and "outboxes" are electronic. Employees no longer have to go "down the hall" to the computer lab to complete a CBT tutorial - they have a computer right on their desk and it may be networked to all the other computers. Along with this shift in hardware availability, more powerful software programs have evolved - one of which is Electronic Performance Support Systems.

Electronic Performance Support Systems (EPSS) are integrated computer-based systems that provide access to information and training. The categories within an EPSS may include reference databases, advice, online help, computer applications, productivity software, and training (Raybould, 1990). The goal of an EPSS is to enable people with limited experience on computers to perform as if they knew what they were doing by providing all of the resources, training, and help they need at their fingertips (Gery, 1991).

Carr (1992) outlines several basic roles that a well-designed EPSS can perform:

- It can act as a librarian. In this role, it helps the performer find, organize and interpret the information required to carry out a task.
- It can function as an advisor. It embodies and shares some specialized expertise that the performer needs to carry out the task.
- It can be an instructor. In this role, it trains the performer in some aspect of the work to be done, just as the advisor role is closest to that of an expert system, the instructor role is an outgrowth of computer-based training (CBT).
- It can serve as a doer. When in this role, it does the work with or without assistance from the human performer (p. 44).

Through these roles, employees are supported on the job with information and training "where they need it, when they need it, in the form most useful to them" (Carr, 1992, p. 44).

CBT vs. EPSS

An EPSS differs from computer-based training (CBT) in many ways. With CBT, the training is often available only by appointment in the computer lab or similar facility. In addition, CBT courses are generally conducted **prior** to a person's need. For example, an employee may attend a CBT session on how to use spreadsheets in anticipation of new job responsibilities. A problem with this approach is that, by the time the employee needs the new skill, a large percentage of the knowledge and skill will be lost on the "forgetting curve." The training component of an EPSS, however, is integrated into the employee's desktop system, along with the spreadsheets, databases, applications, etc. With an EPSS, the training is available **when** the learner needs it, reducing the

problems of retention between training and application.

Another difference between a CBT program and an EPSS is the structure. CBT lessons are generally structured in a hierarchical manner. Either the program branches the learners based on their performances, or students navigate through a series of menus to access the lesson they want. In both cases, interconnections between lesson components are limited. The structure of an EPSS, however, is built on multiple access routes and hyperlinks to other components of the system. This design permits very flexible navigation and information access by users in a nonlinear fashion. Students can access context-sensitive training from their desktops and can easily navigate between EPSS components--i.e., from a spreadsheet to online help to training.

Another difference between CBT and an EPSS is the amount of student monitoring that is available. With CBT, student activity and performance on questions and exercises is generally tracked. This tracking, however, is independent of personnel files and system records. Because an EPSS is more closely related to total employee support, the training component of an EPSS is often monitored and tracked from the system level. The integration of the training component with the system allows context-sensitive advice, information, control, and various types of support.

Benefits of an Electronic Performance Support System

Performance Support Systems are designed to support employees and to allow them to function more effectively as they learn new skills. The electronic systems can dramatically decrease the time required for an employee to master a new position (Geber, 1991). "With performance support information available at the terminal, less experienced people, with less formal training, can provide a high level of service to your customers" (Braasch, 1990, p. 23). The following benefits of EPSS technology were listed by Stone and Villachica (1993, p. 5):

- Decreases training time (20% to 50%)
- Decreases training delivery travel & personnel costs (30% to 100%)
- Increases retention (16%)
- Decreases paper documentation (33%)
- Decreases documentation reading time (20% to 40%)
- Increases productivity (25%)
- Empowers employees with the tools they need to be productive

Performance support systems can also improve the quality of products and the morale of the employees (Legent, 1993). The quality improves because the workers have ready access to support and training. With this access, the employees need less supervision and are likely to provide better service and produce better products. Employee morale improves because people are motivated with increased confidence and pride in their work.

Design of a Performance Support System

There are few established design guidelines for EPSS development. One problem is that the systems are very diverse in their applications and components (Lemmons, 1991). For example, the structure of an EPSS for trouble-shooting a helicopter may be quite different from an EPSS for an office secretary because the needs of the users vary. The following general principles, however, can be presented:

Avoid merely transferring text from paper to a computer screen. "For the system to improve performance, information must be restructured into its most usable form" (Legent, 1993). In most cases, the restructuring results in a decrease in the amount of text because the information is better organized (Raybould, 1990).

Allow multiple retrieval techniques. The user interface of performance systems should enable users to access information quickly through a variety of avenues. For example, a hierarchical menu structure may be complemented by an interactive system map and keyword searches.

Provide visual aids to inform users of their location. Maps, tables, and titles can help users ascertain their position in a system and minimize disorientation (Whiteside & Whiteside, 1992).

Employ instructional design expertise. One of the best ways to ensure that sound design principles are incorporated into an EPSS is to develop it with an instructional designer on the team (Cluskey, 1992).

Case Study

Analysis

Analysis & Technology was presented a requirement to provide support for intermediate level (I-level) maintenance technicians of the A/N37U-1 Mine Clearing Set. There are a small number of technicians and most are not computer literate; therefore, the support equipment needed to be easy to use and be accessible on the shop floor. Additionally, the system was required to provide electronic access to technical documentation (using current electronic versions), training on maintenance procedures as needed to perform each job, access to illustrated parts break-down information, and a job aid to guide performance on an "as required" basis.

The recommended solution was to design and develop a performance support system that integrates all of the required components in a moveable ruggedized cart. The following sections describe the approach taken.

Design

The user interface design is an intuitive icon-based approach that supports easy access to all elements of the system. Within each element, a hypermedia approach was used to link related information. The hyperlinking capabilities of the PSS provide alternate access to information that can help the maintenance technician perform his job efficiently. From each step of a procedure, a user can access the specific technical information on the current step form the technical manual, Illustrated Parts Breakdown information, or a training lesson on how to perform the procedure. Once in a procedure, associated notes, cautions and warnings are provided with audio narration of the precaution, along with accompanying text on the screen, and a prompt to view the technical manual when necessary. This avoids the traditional hierarchical computer-based training approach and allows the system user to access only the needed information at the required time during job performance.

The main menu (Figure 1) offers three main system components for the user - Job Aid, Training, and the Illustrated Parts Breakdown. Other selections from the main menu include a How To Use System Tutorial, Help and Exit. Clearly defined icons on the left side of the screen depict each selection.

The main system component, the job aid, is designed to assist the I-level maintenance technician as he performs procedures associated with component assembly repair, maintenance, and troubleshooting of the equipment. The job aid graphically depicts each procedure. If the maintenance technician knows that a specific subassembly is malfunctioning, he may select that subassembly from the submenu.

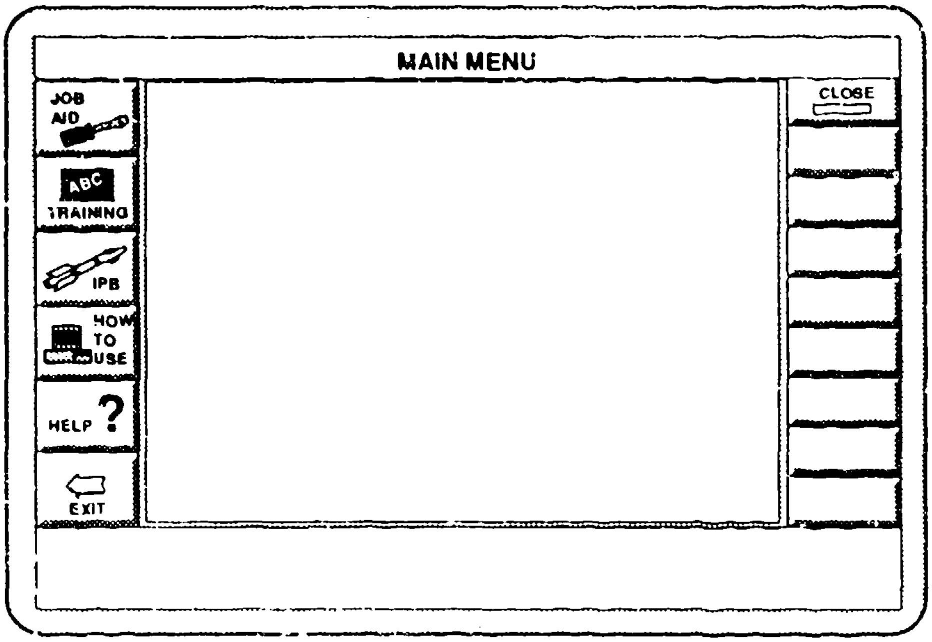


Figure 1.

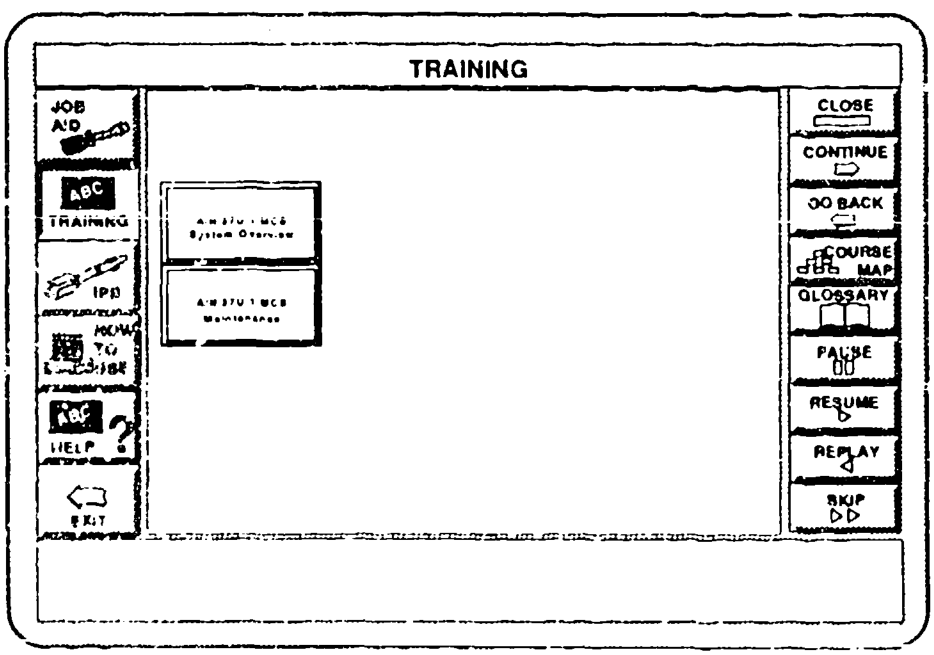


Figure 2.

Once in the job aid, navigation icons illuminate on the right side of the screen offering different options to the user. A tech manual icon provides instant access to the technical manual at the exact location of the step being performed. A glossary is available and the print function will print the job aid flow diagram or the technical manual information, if necessary. When the video icon is selected, video is played that shows the step being performed by a technician. When video is being played, video control options illuminate so that the user can pause, resume, replay or skip a video segment.

The second major system component is training. Training may be selected in two ways in the PSS system. When training is selected from the main menu (Figure 2), training is offered in a traditional CBT (computer-based training) format. The user can select lessons from a submenu and proceed through the desired lessons. When training is selected from within a step of the job-aid, a walk-through of that step of the procedure is depicted.

Illustrated Parts Breakdown (IPB)

Access to parts information is at the heart of the illustrated parts breakdown component. When the IPB is selected from the main menu, a submenu of the equipment assemblies is presented. When the assembly or subassembly is chosen, an Autocad drawing of the assembly appears (Figure 3). The IPB may also be accessed from a step of a job aid procedure. In this case the IPB icon hyperlinks the user to the assembly associated with the step. The navigation icons on the right side of the screen allow the user to manipulate the drawing by panning right or left, moving the image up or down, and zooming in or out. The textual IPB information (Government Standards, vendor, part numbers, descriptions, attaching parts, and SM&R codes) required to order parts appears in the bottom portion of the screen. This textual information, which is presented in the exact hierarchical format of the technical manual, automatically links to the part numbers depicted on the drawing or the text may be scrolled to access a different part. In addition, a specific part may be accessed through the use of a keypad.

Hardware/Software Configuration

The hardware configuration is a DELL 433 DE 33-MHz CPU with 12 MB RAM, utilizing an 80486 microprocessor with a 32-bit, EISA data bus, coupled with a 32-bit, EISA SCSI hard drive controller. Additional components include:

- Elographics Intelli-Touch Surface Acoustic Wave Touchscreen
- Sony LDP 1450 Lasermax Laservision Videodisc Player
- Hard Disk Drives (Internal and Removable)
- SuperVideo Windows Digital Video/Overlay Card
- SuperVideo Windows VGA Daughter Board Accelerated Graphics Card
- Mitsubishi Diamond Scan Monitor
- SoundBlaster Pro Audio Card and Personal Power Speakers
- Hewlett Packard Laserjet IIIP Printer
- Mobile Cart/Stand

IconAuthor authoring software was used to develop the PSS shell, with specific utilities written in the C language to support specific functionalities. The software operating environment is suitable for Microsoft Windows 3.1 with MS-DOS 5.0 (or better).

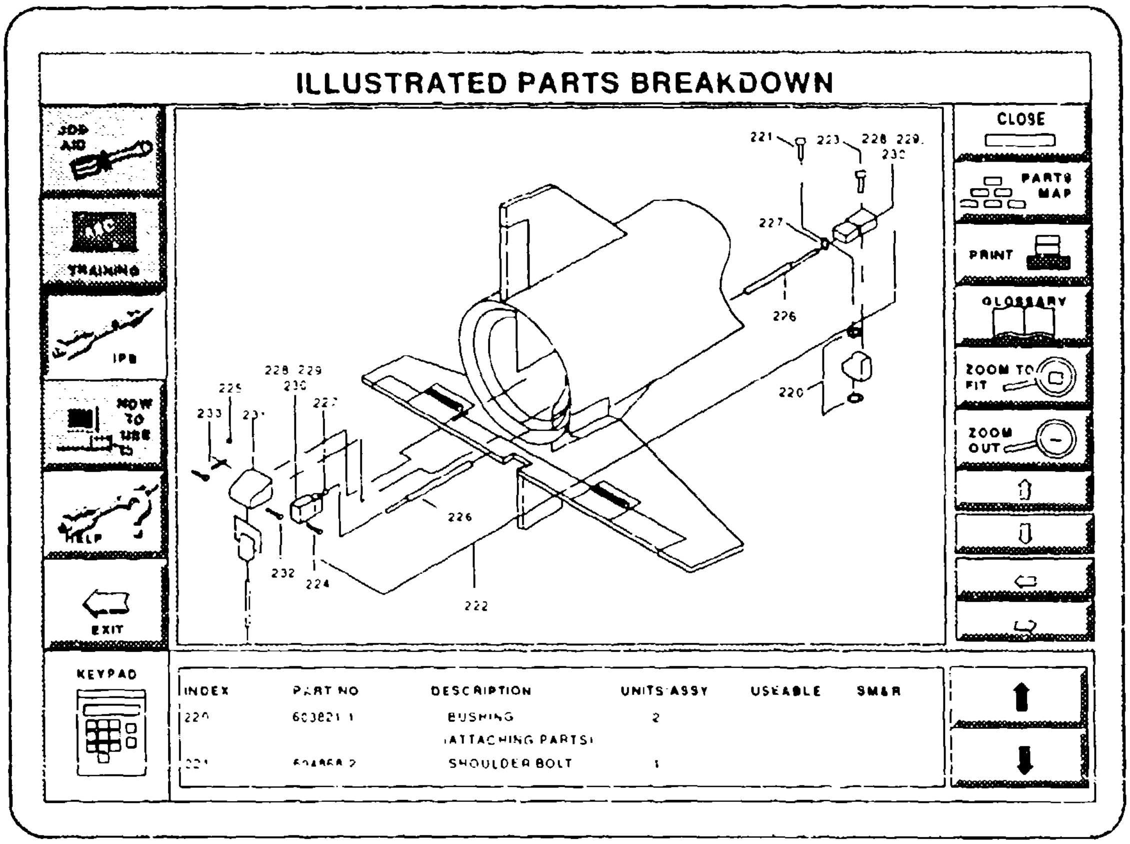


Figure 3.

Development

The PSS development spanned a twelve month period. The development process was aided by the fact that all technical manual information was provided in electronic form. In addition, the Autocad drawings were provided at project start. The development followed the traditional CBT development process, with emphasis and additional time applied to the development of the user interface and the hyperlinking options of the design. A rapid prototype of the user interface was developed by the third month of the development time frame. This allowed designers and subject matter experts to "play" with the design and make improvements early in the process. The team involved in the process consisted of three instructional designers, two programmers, a subject matter expert, two graphic artists, a word processor, an editor, and a project manager. In addition, expert government subject matter experts provided excellent input. Having the subject matter expertise available during the formative evaluation of the product was a key factor in accomplishing the development process in a relatively short time frame.

Implementation

Implementation is scheduled for fall of 1993. Implementation results and lessons learned will be presented at the 15th I/ITSEC conference.

References

Braasch, B. (1990). Performance support systems helping business. *Computing Canada*. 16(4), 23.

Carr, C. (1992). Performance support systems: A new horizon for expert systems. *AI Expert*. 7(5), 44.

Cluskey, M. (1992). The development and implementation of a structured on-the-job training model in one health care foods service operation. Dissertation Abstracts AAD92-20489. Oregon State University.

Fratini, R. (1993). Performance support tools in action: Operational processes. Proceedings of Interactive '93 Conference. May 3-6, Anaheim, California.

Geber, B. (1991). HELP! The rise of performance support systems *Training*, 28, 23 29.

Gery, G. J. (1991). *Electronic performance support systems*. Boston, MA: Weingarten Publications, Inc.

Legent Corporation & Ziff Institute (1993). *Why? Performance support*. Columbus, OH: Legent Services.

Lemmons, L. J. (1991). PSS design: Getting past that first step. *CBT Directions*.4(2), 32-33.

Raybould, R. (1990). Solving human performance problems with computers. *Performance and Instruction*, 29, 4-14.

Stone, D. L., & Villachica, S. (1993). Improving performance with Performance support systems. Proceedings of Interactive '93 Conference. May 3-6, Anaheim, California.

AUTOMATED TOOLS FOR ICON-BASED AUTHORING ENVIRONMENTS: TRAINING DEVELOPMENT AND DELIVERY

ABSTRACT

Icon-based interfaces to authoring systems are a productive and creative new way of designing and developing interactive training courseware. Icon-based tools allow courseware designers (CDs) to create, deliver and maintain computer based training (CBT) courseware, without programming assistance. Lessons are created by selecting icons that represent various types of lesson components, such as graphics, text, pauses, or menus. Custom made icons for frequently used courseware strategies can also be created as "objects" available to all CDs. By arranging these icons, the CD creates a graphic flowchart of a lesson. Through the use of versatile editing environments, often found in graphical user interfaces such as Microsoft Windows and X-Windows, creating and manipulating a lesson is made simple.

After investigating the features of an available icon-based authoring interface, Paramax created a software tool to support flowchart and storyboard development by instructional designers. As part of our courseware productivity IR&D we have developed a tool that parses icons to create an ASCII delimited database file containing all data required by MIL-STD 1379D. That ASCII file is imported into a COTS software application for form generation to create storyboards that the customer can review on screen or print for hard copy review. When the storyboards are approved and appropriate comments are incorporated, preliminary source code is compiled directly from the storyboards.

This paper describes how icon-based authoring interfaces and support tools can increase productivity and configuration control. Advantages of icon-based authoring interfaces and future directions for courseware productivity IR&D efforts are outlined.

About the Author

Ingo Ellerbrock is a member of the instructional staff with 9 years of instructional experience with the Strategic Weapons systems aboard Nuclear Submarines. For the past 5 years he has devoted his efforts towards computer based training. Mr. Ellerbrock was the principle investigator for our courseware productivity IR&D effort. He has earned a B.S. degree in Education at Southern Illinois University and is currently working towards his MS degree in Computer Science.

AUTOMATED TOOLS FOR ICON-BASED AUTHORING ENVIRONMENTS TRAINING DEVELOPMENT AND DELIVERY

Ingo B. Ellerbrock
Unisys Corporation Government Systems Group
Reston, Virginia

INTRODUCTION

Computer Based Training (CBT) has matured to the point that DoD procurements are soliciting large scale (1000+ hours) CBT and Interactive Courseware (ICW) development programs. Most of these large scale efforts have had aggressive development schedules and some were linked to the development of tactical systems. To effectively support future large scale requirements, the Unisys Information Systems Training Department of Unisys Government Systems Group determined that we needed to identify tools and processes to increase CBT development productivity. Although, we had considerable experience and proficiency in code-based authoring systems, we decided to evaluate available authoring systems designed to maximize designer and author productivity.

PROJECT OBJECTIVES

This paper describes a research project by Unisys to increase productivity in the development of large scale CBT programs. Project objectives include:

- Selecting an authoring environment that supported development of productivity tools

- Complying with the letter and spirit of MIL-STD-1379D

- Creating productivity tools for flowcharts and storyboards.

ICON-BASED AUTHORING SYSTEMS

Until recently, most robust CBT authoring systems required that each lesson be coded line by line. Some languages were so complex that they required the skills of a computer programmer on the development team. Using paper storyboards created by instructional designers as a specification, frame content and presentation effects were coded using a text-based, syntax-sensitive authoring language. Even today, many CBT lessons are developed this way. Due to the high-level design specification and programming expertise required, this process can be extremely labor intensive.

Icon-Based interfaces to authoring are one of the most revolutionary changes to occur in the evolution of CBT development. In less than ten years, the interaction between developer and computer has changed from terse, syntax-sensitive ASCII-based programming languages, to the now familiar Windows, Icons, Menus, and pointing device interface. This advancement has increased the accessibility and usability of computer based systems to non-programming professionals and created a commonality of use across platforms. This commonality can contribute to decreasing the learning curve when moving from one authoring environment to another.

An example of an Icon-based authoring system is depicted in Figure 1. An icon-based system offers a library of icons that represent groupings of computer code. Each icon represents a unique object, structure, or process. For example, a videodisc icon represents code for the control of a video sequence in which the videodisc player starts and stops at specific frames, plays in slow motion, or displays a still frame. By grouping icons in a particular fashion, computer-based lessons are formed.

Advantages to the use of icon-based authoring systems in large-scale, multidisciplinary team environments include the following:

- Icons are intuitive, easy to learn, and often already familiar from previous experience.

Most communication takes place through the exchange of signs[2]. Icons came from visual semiotics, the study of visual signs. Icons communicate by virtue of their inherent physical characteristics that make them look like the objects to which they refer. For example, an icon that looks like a movie camera is intended to represent a sequence of an animation.

- Icons promote reusability

Reusability of courseware code can be a key to improving courseware development productivity, quality, and consistency. The reuse of courseware components increases the courseware developer's productivity by allowing the developer to write fewer total symbols in the development of a system, and to spend less time in the process of organizing those symbols. Research efforts in software development processes has shown that reuse is enhanced by icon-interfaces specifically because they hide the details of implementation and raise the level of discourse to the problem domain level rather than the implementation level[1].

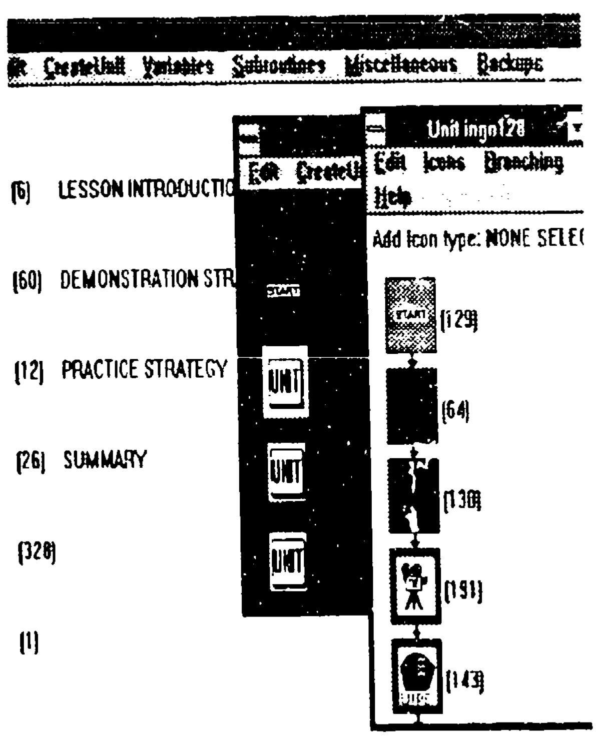


Figure 1.

Looking at icons as blocks of lesson code, and the ability to easily move, cut, and copy them provides a positive, object-oriented environment for courseware reuse and design control.

- Visual Formalisms

Visual formalisms (Figure 1) are diagrammatic displays with well-defined semantics for expressing relations[3]. Examples of commonly used Visual formalisms are tables, graphs, plots, and panels. A Visual formalism is intended to help developers organize and present an entire complex application, or a significant piece of one. Visual formalism, in terms of icon-based authoring systems, is the presentation of icons logically connected to one another, giving the developer an instant broader sense (big picture) of an entire lesson or large sections of one. This allows exposition of details at various levels.

- Ease of Use

One obstacle to a broader acceptance and implementation of CBT has been the perceived high cost of courseware development[4]. The large number of labor hours required for courseware development drives the cost of CBT. Icon interfaces to authoring provide a higher level interface than the line-by-line programming of the past. This allows individuals with few programming skills to create sophisticated, complex lessons without the need to code. The result is a broadening of the skill mix that can be used for courseware development and a reduction in cost due to a reduction of programmer time required to create computer-based lessons.

SELECTING AN AUTHORING ENVIRONMENT

Although the authoring system selection criteria was focused on productivity, other development and delivery factors were also considered. The following characteristics were used in the selection process:

- Multiple platform development and delivery (DOS/UNIX)

- Dual screen capabilities

- Ease of interfacing third-party functions
- Support for electronic storyboarding
- Support for rapid prototyping of courseware
- Ability to import word-processor based text files
- Ability to import and export graphic files from/to external paint, drawing, and animation programs
- Instructional strategy libraries or templates

The Training Icon Environment (TIE™), developed by Global Information System Technology, was chosen as the authoring system for exploring future courseware development productivity issues. TIE was selected for the following reasons:

- Icon-based interface
- Supports custom icons
- Supports strategies and templates
- Access to underlying ICW ASCII code
- Supports the requirements of large CBT projects.
- Supports access to coded language for more complicated presentations

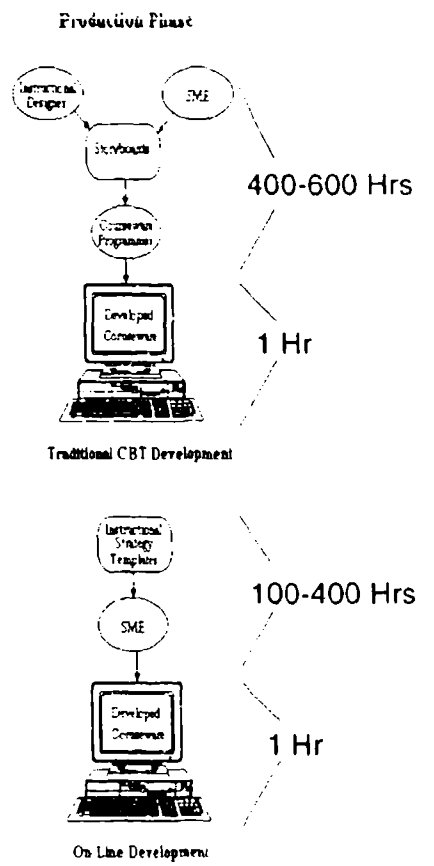


Figure 2.

INSTRUCTIONAL STRATEGY TEMPLATES

After investigating various features of icon-based authoring environments, it was determined that the use of previously developed courseware reduced development time and increased courseware reliability. Reusable instructional strategy templates were developed which allow individuals not trained in instructional design, to create courseware. Instructional Strategy templates are well-designed groupings of content-free icons which define the basic interactions contained in a segment of a lesson, as well as the lessons higher-level organization and sequencing.

By enforcing use of predefined instructional strategies, developers spend more time on focusing what is to be taught than on how it is to be implemented. In addition, the use of well-designed strategy templates allows, the Subject Matter Expert (SME) to develop materials on-line with much the same quality as an experienced courseware developer.

Additional productivity gains can be made by altering the traditional path to developing courseware, reducing the time spent by instructional designers and courseware developers. These individuals would now spend most of their time during the initial design and prototype stage in the development of project-specific instructional strategy templates. Subject Matter Experts, using these templates, could now develop courseware on-line and carry most of the responsibility of development during the production stage of a project. (Figure 2).

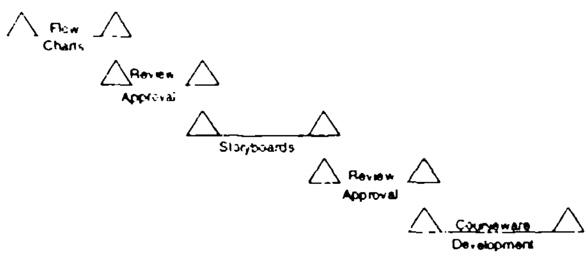
ISSUES

Our project needed to deal with two loosely related issues:

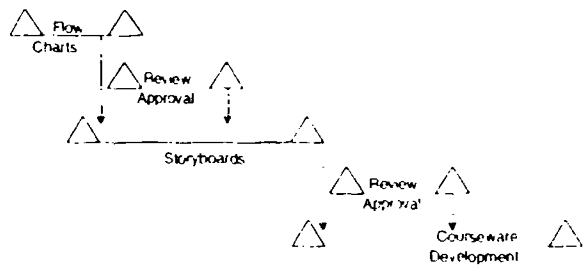
- Process and Contract Deliverable Requirements Lists (CDRLs) in recent procurements dictated that flow charts and storyboards be reviewed and approved serially prior to the start of courseware development (defined in the procurements as "coding the lesson")

- Productivity in terms of development ratio had to be increased to meet large quantity CBT deliveries on a 24-36 month schedule.

The Data Item Descriptions (DIDs) that encompassed flow charts and storyboards and specified delivery requirements dictated the following development cycle.



Significant productivity gains could be made, in terms of both level of effort and calendar time, if the following development cycle could be supported:



We made a subtle distinction between this approach and a prototyping approach. Our intent was to use the authoring system features to develop flow charts and storyboards with the subsequent lesson code as a by-product. This differs from developing the lesson as a prototype without approved flow charts or storyboards. The distinction was an important one to meet the requirements of MIL-STD-1379D and the associated documents typically specified in procurements.

The resulting goal was strict conformance to MIL-STD-1379D without losing the productivity gains of an on-line development concept.

Flowcharts and storyboards created as a specification for lesson development must be reviewed and approved by the customer. If developed on-line within the authoring system, they can also be used to manage and control design format and consistency for large scale efforts that required many designers, developers, and SMEs. Development of reusable templates to provide this management control is inherently supported by icon-based interfaces.

PRODUCTIVITY TOOLS

After evaluating the available features of the TIE authoring environment, we concluded that the biggest productivity gains could be realized by focusing on the lesson flow charts and storyboards requirement. Of these two processes, we felt the storyboard process was the bigger challenge with higher potential productivity gains. Therefore, our first goal was to create a software tool that takes advantage of the icon-based authoring features to support storyboard development. When storyboards designed and created using the authoring environment were approved by the customer, code representing approved storyboards would be compiled into Accord TUTOR code. This code compilation from the higher level interface also reduces production effort associated with coding errors and debugging.

Design Considerations

The authoring system allows users the flexibility to position icons in various groupings or structures. As part of our design for the instructional strategy templates we needed to define what a storyboard or frame is. An icon

structure was designed which required developers to adhere to design strategies implemented in strategy templates.

Figure 3 shows the relationship of the icon structure to our instructional strategy template for courses, lessons, and storyboards. Lessons consist of one or more modules. Each module consists of multiple units. A unit is equivalent to a single frame in a CBT presentation. The design for each unit is documented on a single storyboard. Within each unit, the developer defines the content and interaction elements of that unique CBT frame on the training requirement and approved lesson flow diagrams. We have defined these frame elements at the unit level with an icon structure. This structure creates a content and interaction checklist which aids the developer in developing each frame. Using simple point and click techniques, the developer can specify frame parameters using the icons and their associated dialogue boxes.

The FAST Tool

The next step in our courseware productivity IR&D, we developed the Facility for Authoring Storyboards in TIE (FAST) tool. Figure 4 illustrates the relationship of the FAST tool to TIE and the storyboard design process.

TIE creates three groups of files during courseware development sessions: ASCII TIE Code files, ASCII TUTOR Source files, and the Compiled Binary Run file (Figure 5). The ASCII TUTOR Source files, derived from ASCII TIE Code files, are the files which, when compiled, create the Binary Run file. The ASCII TIE Code files contain data which is only used during a TIE editing session. They are the initial repository for courseware data. They are also the data that the FAST tool uses to create storyboards. ASCII TIE Code files are broken up into three categories of files: The Icon Navigation File, the Icon Display Text Files, and the Icon Description Files (Figure 6). The Icon

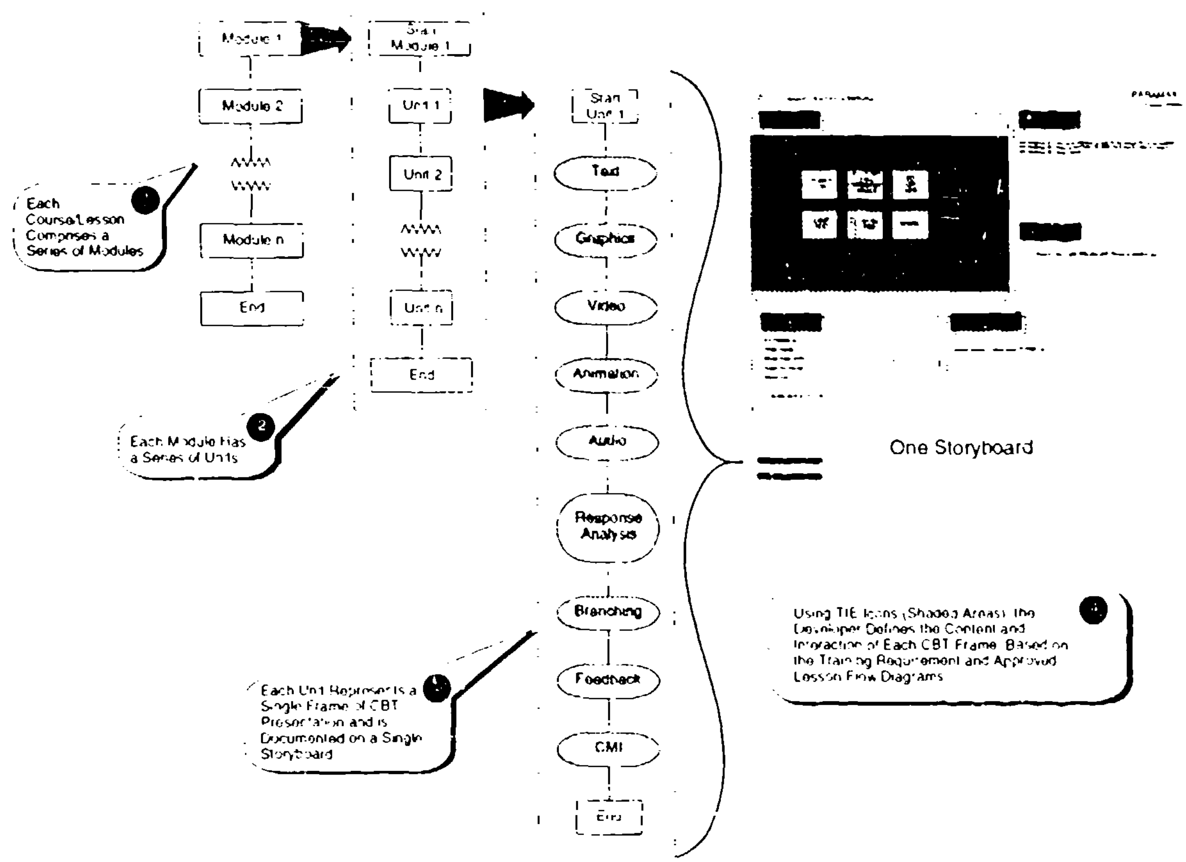


Figure 3.

Navigation file is a textual representation of the icon path. It is essentially the road map for determining the sequencing of icons. The Icon Display Text files contain the text displayed on the screen, its position placed on the screen and its font and size. The Icon description files contain the rest of icon information (e.g., animation file location, touch areas, and graphic file location). Our FAST tool, using the Icon Navigation file, parses the ASCII TIE Code files to create an ASCII delimited database file containing data required by MIL-STD-1379D.

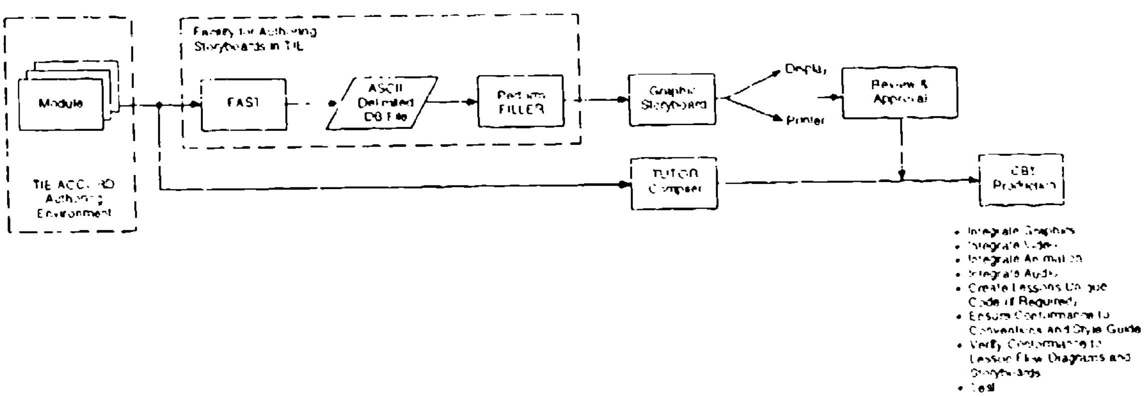


Figure 4.
One record of the ASCII delimited database contains information representing one storyboard.

After the Icons have been parsed and entered into the database, the database is then imported into a "forms" software application package for storyboard generation. For our project we selected Delrina's PERFORM Pro+ because we had experience with the product and it supported TIE graphics (GIF). It also is very flexible which could be used for follow on research and development projects. Figure 7 is a sample output from our FAST tool.

With the conceptual end product shown in Figure 7 in mind, the designer/developer can use the features of the authoring system to layout or each frame. The expected productivity gains were:

- Links and branches were automatically established and managed by placement of lesson element icons

- File names as placeholders for graphics are established. Sketches could be scanned into the storyboard at the same time the requirement is being sent to the graphics support for development and control as part of the project graphics database.

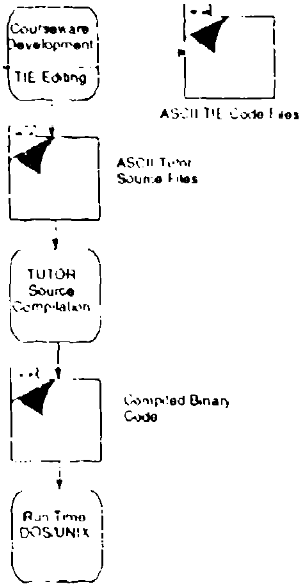


Figure 5.

Storyboard reviews, both internal and customer, can take place on screen or print for hard copy review.

When storyboards are approved and appropriate comments are incorporated, the lesson goes into CBT production for the following activities:

- Create and integrate graphics
- Acquire and integrate video
- Create and integrate animation
- Create and integrate audio
- Create lesson unique code (if required)
- Ensure conformance to conventions and style guide
- Verify conformance to lesson flow diagrams and storyboards
- Test and debug

SUMMARY

The FAST tool was able to create storyboards from design data that was input directly into the authoring system. We were successful generating storyboards for both new lesson development and from lessons that already existed. This was achievable because of the open nature of the authoring environment. Access to underlying ASCII files allowed us to extract data needed for storyboard development. To date, the product and approach has not been applied to an actual CBT development program. Therefore, no productivity data are available.

Although our efforts were technically successful, one disappointing discovery was the effort required to develop FAST-like tools for other authoring systems and environments. Authoring systems are structured and organized differently. Because of this, a substantial amount of rework would be required for conversion of the FAST tool to other authoring systems.

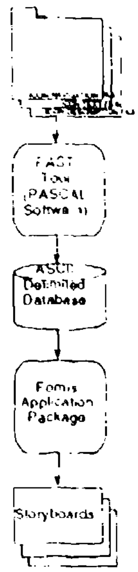


Figure 6.

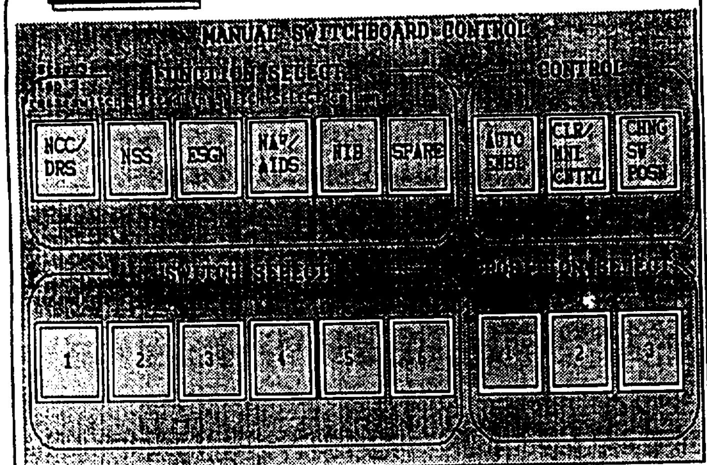
FUTURE CONSIDERATIONS

One possible solution to lower cost of CBT development is an open standard for ICW data.

The seven-layer open systems interconnect (OSI) model has been a catalyst for the evolution of conventional hardware and software to an open environment. Data are now being shared from heterogeneous applications on a wide range of hardware configurations. Interactive courseware development would benefit in terms of productivity and management

PARAMAX
A Unisys Company

Display Screen



Feedback

WA feedback # 1: YOU HAVE TOUCHED AN INCORRECT AREA. PLEASE TOUCH ANY PART OF THIS YELLOW AREA TO CONTINUE.
WA feedback # 2: YOU HAVE TOUCHED THE WRONG AREA FOR THE SECOND TIME. Refer to SSP OO 57650, NOP 90A.3. Step .
WA feedback # 3: That was your third try. Please touch the red area and then FUNCTION SELECT ESGN to continue.

Branching info

TOUCH AREA: (276,153),(326,101) Branch to Storyboard 21

Visual Elements

IVD FRAMES (S) ______
TIME PAUSE ______
ERASE-FADE-WIPE ______
QUESTION FRAME ______
ANIMATION ______

** BLANK DEFAULTS TO NO **

Special Instructions

When AUTO ENBL is pressed it illuminates white then back to green.

Figure 7

if similar standardized data exchange levels and formats were established.

If all authoring systems were required to exchange certain key data elements, third party utilities could be developed to support courseware design, development, data management, and configuration management (Figure 8).

To extend the automated storyboard and flowcharting approach to multiple authoring environments would require such a data exchange standard. This would be a first step in creating an open systems environment for interactive courseware development and delivery. Some elements of an open environment are being realized through Windows utilities for object link and embedded (OLE) and dynamic data exchange (DDE). However, there are no efforts specifically targeting CBT and ICW requirements, especially in a DoD environment.

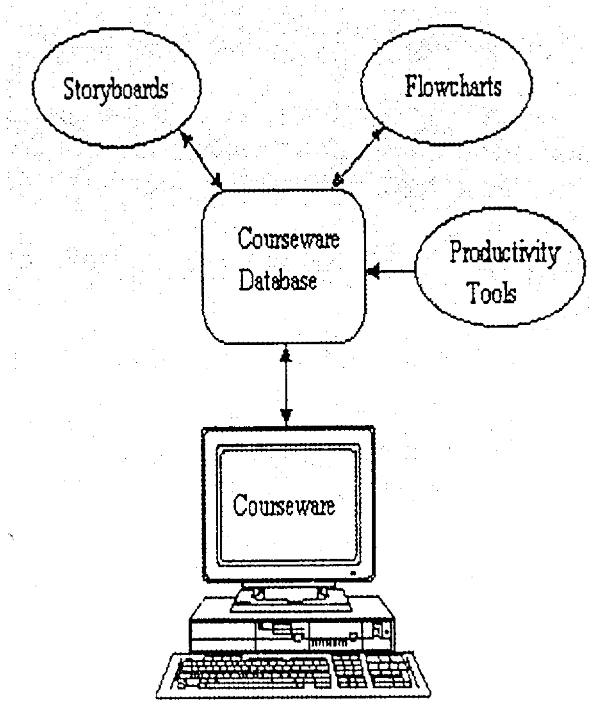


Figure 8.

REFERENCES

1. Biggerstaff, T.J., & Perlis, A.J. Software Reusability ACM Press New York, New York 1989.

2. Chen, T.T., Brandt, L.M., Barbee, D.E., & Lorenc, D.R. The Global Guide to the World of CBT. GLOBAL Information Systems Technology, Inc., 1993, 4-1 - 4-26.

3. Gane, C. Computer-Aided Software Engineering: The Methodologies, the Products, and the Future. Englewood Ciffs, New Jersey, 1990.

4. Johnson, J.A., Nardi, B.A., Zarmer, C.L., & Miller, J.R. ACE: Building Interactive Graphical Applications. In Communications of the Association for Computing Machinery, 36(4) April 1993, 41-53.

5. Marcus, A. Human Communications Issues in Advanced UIs. In Communications of the Association for Computing Machinery, 36(4) April 1993, 101-109.

TAKING ADVANTAGE OF LOW-COST COTS SOFTWARE FOR THE DEVELOPMENT OF TRAINING MANAGEMENT SHELLS

Ellen E. Shay and Linda L. Terlecki
Loral WDL
Hanover, Maryland

ABSTRACT

In the past few years, a wide variety of powerful, low-cost commercial off-the-shelf (COTS) software packages have been released, allowing users to build complex applications using minimal programming skills. These packages have made graphical user interfaces particularly simple to develop by providing robust on-line tools and support features. This allows applications to be quickly and easily prototyped for early user involvement, better user understanding, and overall proof of concept. Developers can concentrate on the requirements and design of an application, spending time on the "look and feel" of the application instead of the "how" because the how has been simplified.

Our requirements were to build a training system management shell that provided student logon, access to course materials, management of student data, and course evaluation data reporting. This shell was part of an overall effort to produce a general-use, Multimedia Personal Computer (MPC)-compliant platform that was also to be used for language enrichment materials. This platform included a specified set of hardware and COTS software. We analyzed the given set of software tools, then developed a strategy to enhance the overall training product by providing a training system management shell for a minimal investment. It was determined that the best strategy would be the use of the built-in capabilities of the provided COTS software. The training system management shell was developed with a minimal use of traditional software development procedures, focusing only on the essentials for successful user management in the specific environment.

We found this approach to be appropriate when it is necessary to enhance existing student management and course evaluation capabilities, integrate courses from different sources, minimize time/resources spent on non-instructional aspects of a project, accommodate a short development schedule, and/or utilize resources whose skill level and/or availability won't allow a traditional approch to development.

ABOUT THE AUTHORS

Ellen Shay is currently a Training System Engineer at Loral WDL. She has integrated hardware, software, and courseware for several training systems. Prior to joining Loral in 1984, she was employed by McDonnell Aircraft Company as a Logistic Element Manager for Training and an Instructional Designer. Ms. Shay holds an MS in Media Education from Florida State University and a BA in Computer Science from the University of Florida.

Linda Terlecki is currently the Project Engineer for the Job Skills Education Program, a basic occupational learning program at Loral WDL. She was the lead system engineer for the development of several training delivery systems deployed world-wide. Ms. Terlecki has been involved with the design and development of computer-based training systems since 1980. Prior to joining Loral, Ms. Terlecki was a software engineer with the Hazeltine Corporation. Ms. Terlecki holds a BA in Political Science from Dickinson College in Carlisle, PA.

TAKING ADVANTAGE OF LOW-COST COTS SOFTWARE FOR THE DEVELOPMENT OF TRAINING MANAGEMENT SHELLS

Ellen E. Shay and Linda L. Terlecki
Loral WDL
Hanover, Maryland

INTRODUCTION

Until recently, the development of computer-managed instruction (CMI) systems required that almost all components be built from scratch, including the database, user interface, and data reporting. This effort required a lot of specialized resources. In addition, the user interface wasn't particularly intuitive, given that development tools were unavailable and the project personnel were busy developing the database. Fortunately, the software industry has produced inexpensive, powerful applications that have revolutionized software development. As developers and users, we can now go to the local computer store and purchase much of the functionality our projects require. We may then spend our time customizing the product, making it easier for our customers to use.

This paper examines the current capabilities of these commercial off-the-shelf software (COTS) packages and how they were used to develop a training management shell.

USING COTS PACKAGES

Capabilities of COTS Packages

Require Minimal Programming - Recent COTS releases allow users to build complex applications using minimal programming skills. Spreadsheets are being turned into custom financial analysis programs and databases are becoming sophisticated inventory and tracking systems. COTS packages include tools that allow users to develop elaborate custom applications with minimal help from programmers. Tasks that used to require a lot of programming may now be provided as an option on a menu or button bar.

On-Line Tools and Support Features - Robust on-line tools and support features have made the user interface particularly simple to develop and customize. Frequently-used functions are easy to automate. These features assist and simplify the translation of paper-and-pencil designs into polished applications with all the standard Windows features, including buttons, pull-down lists, and graphic logos. Some packages have "Wizards" or "Writers" for screen forms and reports, making it easy to quickly generate a form or report from a query. These features ask a few questions, present options that represent the most common designs, then automatically generate the form or report. Graphical tools are used to draw forms and reports instead of programming. Macro languages allow developers to completely customize the look and operation of the underlying application—even to the point of replacing its menus, screens, and dialogue boxes. The end result is easier-to-use, custom-fit interfaces for users that are indistinguishable from a program written from scratch in a language such as C. Now that COTS packages are moving towards object-orientation, developers and users can select an object in a table, form, or report and set the options associated with that object. This allows the developers and users to make complicated changes to forms or reports very easily.

On-Line Helps - COTS packages now provide extensive on-line helps. Most packages offer context-sensitive help, giving you information specific to the task you are performing at the time you requested the help. Windows-based applications have included the capability to annotate Help files, customizing them. There are also packages that allow

developers to build Help files for applications they build using COTS packages.

Issues in Using COTS Packages

There are some issues that should be considered when using COTS packages. There has been an explosion of software, especially software employing graphical user interfaces. The capabilities of the available software are changing almost as fast as hardware. If the press releases are to be believed, better and/or revised products are always just about to be released. Developers and users find that they must answer the question: Should I use the current version or wait some number of months for the next version or a new package? This question can cause schedule problems if the wrong decision is made. With the rapid increase in the number of COTS packages, developers and users must ensure they are aware of the pitfalls that may occur with beta releases and first versions of new software. If there are problems with the COTS package, your project will be at the mercy of the software company for fixes. You have no control over the timeliness of the fixes. On the other hand, it is their problem to fix and not yours.

Benefits of Using COTS Packages

The benefits of using COTS packages far outweigh any issues concerning COTS packages.

Allows Short Development Schedule - Using COTS packages accommodates a short development schedule. Most of the desired functionality is built-in or easily produced. COTS packages, for the most part, have already been through extensive testing—developers can concentrate on testing the modifications they made.

Concentration on "Look and Feel" vs. "How" - Developers can concentrate on the aspects of their project that would be difficult and/or time-consuming no matter how they were implemented. Developers are able to spend time on the requirements and design of the project because the built-in features of the COTS package sho' :ns the development phase. They can spend their effort on the "look and feel" of their product instead of the "how" because the how has been simplified. The custom application can take advantage of all the high-level facilities provided by the underlying application. For example, an elaborate field-transfer protocol or calculation that might require hundreds of lines of code in a conventional language can be invoked with a single macro language function call.

User Interface is Easier to Develop - The user interface is easier to develop than in the past. Most COTS packages provide interface design tools and macro functions that specifically support user interface development. In particular, Windows applications take advantage of the graphical environment to allow end users and developers to create sophisticated applications with very little programming.

Using the power of the COTS features, developers can prototype applications such as a training management shell. Prototyping allows for early customer/user involvement with better understanding by the user. Developers may also try out different solutions to the requirements, proving their concepts. Once the product is delivered, the customer and/or user can customize and tailor the product, making the product maintainable by the customer.

Benefits End Users - COTS packages also benefit the end users by requiring less training and increasing productivity. First, many users are already familiar with many of the COTS packages on which the customized applications are based. Second, developers use features such as macro languages to hide the rows and columns of spreadsheets and to design document templates for word processing.

There is a wide variety of powerful, low-cost packages that are just as effective as many of the more expensive, complicated packages.

APPLICATION OF COTS TO A TRAINING MANAGEMENT SHELL

Overview of Project

Our main objective to produce a general-use, Multimedia Personal Computer (MPC)-compliant delivery platform, including hardware and software, that was also to be used to deliver language enrichment materials. We analyzed the given toolset, then developed a strategy to enhance the overall product by providing a

training system management shell and data reporting for a minimal investment. However, the training system management shell was not to support an elaborate set of features or user interface.

In addition, we were tasked with converting existing courses that used an outdated interface and were running on outdated technology. The content and design of the courses were to stay the same. The hardware and software platform would change.

Requirements - During the development of the training management shell, the focus was on providing only the essentials for successful user management in the specific environment for the courses. The requirements included:

- User log on
- User progress data collection and backup
- Course evaluation (data analysis)
- Course access
- Access to full capabilities of the COTS packages delivered with the system (e.g., Microsoft Word)

Hardware and Software - The customer specified the hardware and software that was to be used as the delivery platform. The hardware suite, an MPC-compliant PC with interactive video, is defined in Table 1. A comprehensive list of COTS software is listed in Table 2.

Table 1 Hardware Suite

COMPONENT	DESCRIPTION
CPU	486-66MHz CPU with Dual Floppy Drives, 256K Cache, 32MB RAM
Hard Drive	660MB Hard Drive
Input Devices	Keyboard, Mouse
Monitor	19" Multisync Monitor
Media Sources	CD-ROM Drive, Videodisc Player
Graphics Cards	Graphics Adapter, Video Overlay
Audio	Digital Audio Card, Audio Mixer, Audio Amplifier, Audio Speakers

Table 2 Project Software

TYPE	PACKAGES
Operating System	DOS 5.0, Windows 3.1
Utilities	Virex 3.1, Norton Desktop Windows 2.0, Grafplus Screen Print
Authoring	Authorware Professional for Windows 1.1
Font Utilities	All Type, Adobe Type Manager, Kaballah Font, Fontographer
Development	Microsoft SDK C/C++ 7.0, Microsoft Access 1.0
Other	Microsoft Word for Windows 2.0

Rationale for the Approach

There were three factors that affected the approach we used in the development of the training management shell:

- The customer provided a specific suite of hardware and software
- The development schedule was short
- Minimal resources, including staffing, were available for the project

We considered several approaches before deciding to use the built-in capabilities of the provided COTS packages, with minimal coding.

Description of the Process

Overview - The functional requirements were specified during the definition of the user interface. The process was iterative, with overlap between the Analysis and Design phases and the Design and Development phases. The data collection requirements were derived from the reports specified. See Figure 1 for a graphic of the basic process.

Analysis - The development team, working with the customer, used flowcharts to define the concept of operations (CONOP) for the utilization of the system and the desired data analysis reports. (See Figure 2). The following is a brief description of the CONOP:

1. Student selects the icon of the desired language.
2. If the student is already registered, the student enters the course. If not, the student registers.

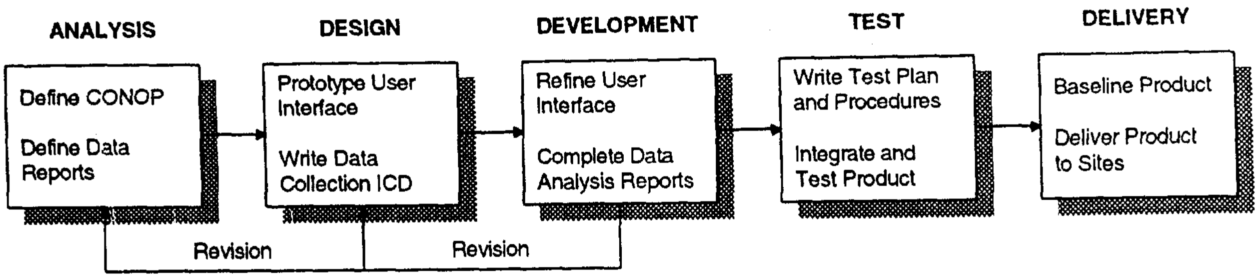


Figure 1 Training System Management Shell Development Process

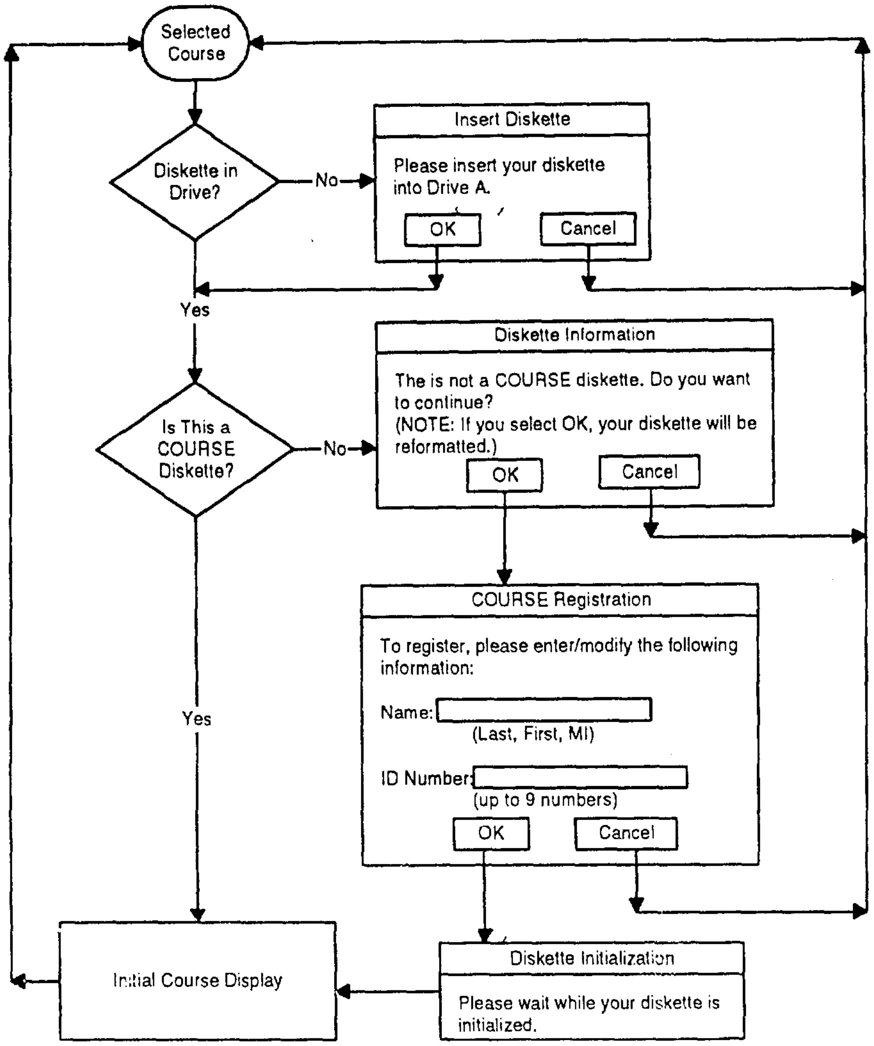


Figure 2 CONOP Process Example

3. Progress data and test results are saved to the student's diskette.
4. When the course is completed, the student sends the diskette to the schoolhouse.
5. The schoolhouse imports the data.
6. The schoolhouse accesses data analysis reports.

Design - The functionality of the user interface and reports was designed using line drawings of the proposed displays. We produced a Project Implementation Plan that included the user interface, reports, and CONOP. The COTS software was then used to prototype and polish the user interface. Design reviews with the customer consisted of

presentations of both the line drawings and on-line prototypes. The customer provided comments on the design, which was then revised for further review. This process resulted in a detailed design for implementation. The courseware and software Interface Control Document (ICD) was written for data structures and data collection. Due to the rapid prototyping, much of the interface for user management was complete by the end of the design phase.

Development - As each part of the shell was prototyped, it was demonstrated to the customer for feedback. The user management interface was refined and the course evaluation (data analysis) reports were completed.

Integration and Test - The Project Implementation Plan was used as the basis for test plans and procedures, making them easy to write and trace back to the requirements. Full functionality, data collection and the user interface were all tested. The COTS software operation did not need to be verified—we only had to ensure the training system management shell was fully integrated with the courseware and worked properly. The applications developed with COTS software were integrated and tested incrementally. The courseware developers and the customer were responsible for testing the converted course. Final integration and test verified that the training system management shell and converted courseware operated smoothly and could transfer data. The time spent during the analysis and design phases and the use of COTS products led to a high return. Integration and test was completed ahead of schedule, with only one anomaly.

Product Delivery - The courseware and software were baselined, loaded on the hardware, then delivered to the sites. The delivery consisted of an integrated system including the training system management shell, courseware, hardware, user manual, and student guide.

Examples of Displays

This section provides some examples of the user interface displays that were developed using COTS packages.

The screen shown in Figure 3 is the Language Enrichment Group. The student selects the course from this display.

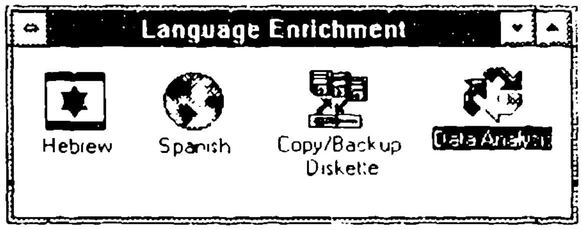


Figure 3 Language Enrichment Group

The screen shown in Figure 4 is used by the student to register for a course.

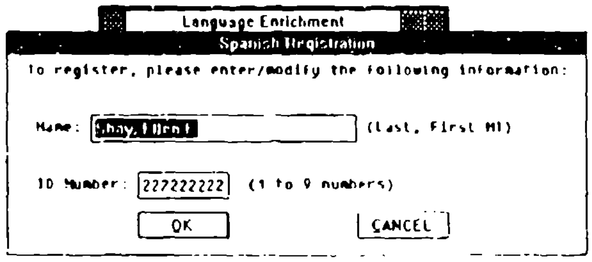


Figure 4 Student Registration

The screen shown in Figure 5 is the Data Analysis Group. The schoolhouse selects the desired data analysis action.

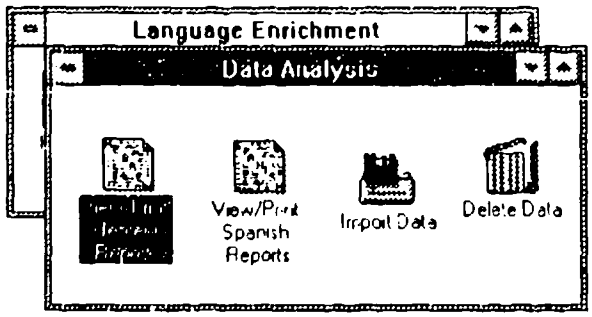


Figure 5 Data Analysis Group

The screen shown in Figure 6 allows the schoolhouse to specify the student(s) to report on.

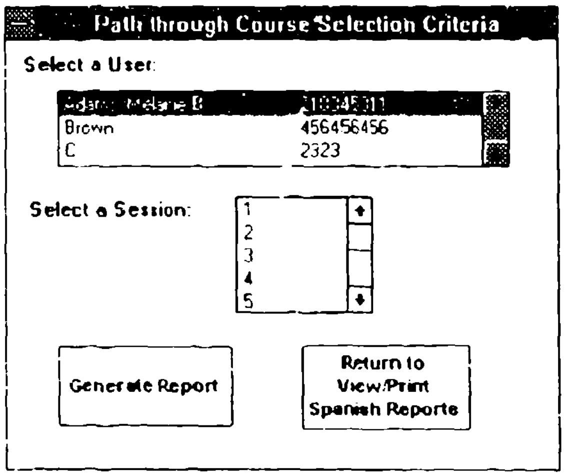


Figure 6 Selection Criteria Screen

The screen shown in Figure 7 is an example of a report.

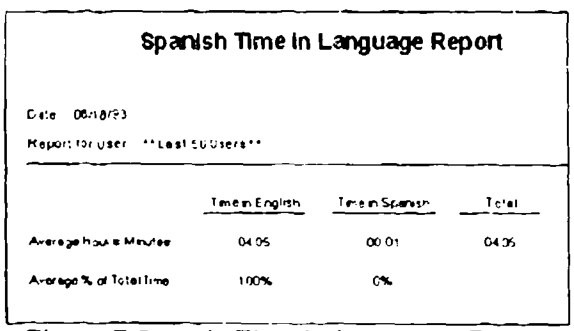


Figure 7 Sample Time in Language Report

SUMMARY

Results of Process

The use of COTS software for the development of the training system management shell produced two major results:

1. The requirements as defined by the customer were met to their satisfaction.
2. The development of the shell was completed four weeks earlier than originally estimated and required less re-work than more traditional development methods.

When to Use This Approach

Using COTS packages for training management shells is appropriate when a project requires:

- Enhancements to student management and/or course evaluation capabilities
- Integration of courses from different sources
- A minimum of time/resources spent on non-instructional aspects

On a more generic level, COTS packages should be considered when any project requires:

- A short development schedule
- That the existing resources be utilized, but the skill level and/or number of available resources will not allow a traditional approach

Benefits

Using COTS packages has several benefits. The product is easy to enhance and maintain and is self-documenting. Developers can concentrate on the requirements and design instead of the mechanics of implementation. Prototyping can be used to ensure customer understanding and to prove concepts. The basic COTS package features may be used to meet most users' needs. End users require less training and are more productive.

Some Final Thoughts

When developing training management shells, do not rule out COTS packages. They can be just as effective and much less expensive than more complicated approaches. Finally, Christine Cornaford of PC Week put it well:

> "I used to program for a living. However, given a choice between solving a problem by writing some code or buying the solution off the shelf, I'd take the store route every time. So should you. With the variety of terrific tools available today, you can spend hundreds of dollars to save thousands more in unnecessary coding. Why not save your programming efforts for the critical pieces of your application development environment and buy the rest?"

REFERENCE

Comaford, C. (1993, February 15). Tools that simplify the GUI environment. PC Week, p. 63.

THE DEVELOPMENT OF THE EMBEDDED TRAINING DECISION-AIDING AND RECOMMENDATION TOOL (ET DART)

ABSTRACT

The development of embedded training (ET) guidelines by the Army Research Institute (Witmer and Knerr, 1991a, 1991b) has afforded military planners a systematic method for making critical training decisions related to embedded training systems and other training alternatives. Recent work at NTSC and STRICOM has attempted to expand the use of the guidelines by providing an automated version of the algorithms used in the model. The current study reviews and incorporates those efforts while continuing to expand the capability of an automated version of the guidelines in the areas of:

- user interface;
- making the terminology generic to all services (where necessary):
- user help and instruction;
- decision documentation; and
- decision-aiding for training media cost analysis.

This paper describes the current and future development of a tool (the ET DART) that will fully support decision-making processes related to embedded training.

AUTHORS

P. Kelly Watson, Ph.D. is an educational psychologist in the Customer Training Systems Department of Lockheed Aeronautical Systems Co. in Marietta, Georgia. He is currently the Principal Investigator for the Total Training Systems Integration IRAD program at Lockheed, under which the ET DART is being developed.

Douglas J. Miannay is a Product Support Analyst Sr. with the Customer Training Systems department at the Lockheed Aeronautical Systems Company. He has been involved in training systems development for various programs including F-22, C-130J, and JPATS. Mr. Miannay holds an Associates Degree in Applied Sciences from the Academy of Aeronautics in New York and is currently pursuing a degree in Computer Science.

Douglas Brashear is a graduate of the American Technological University in Killeen, Texas with a degree in Technical Education. Currently he is the Project Lead for Training Systems Integration in the Lockheed Integrated Support Systems project where he is leading the development effort for TARCS and the Training Effectiveness Model for JPATS.

Richard A. Copeland, Jr. is a systems engineer in the Engineering Management of Training Devices Division of STRICOM. He is currently working on the integrated training systems for the Advanced Field Artillery System and the Future Armored Resupply Vehicle. Prior to STRICOM he was a project engineer at the Naval Training Systems Center in the Air ASW Trainers Branch. Mr. Copeland holds a Bachelor's Degree in Electrical Engineering and a Master's Degree in Simulation Systems from the University of Central Florida. He is a certified Engineer-In-Training and a member of Tau Beta Pi and Eta Kappa Nu.

Nina E. Chatham is currently a post graduate fellow at the Naval Training Systems Center in the Oak Ridge Institute for Science and Education program. She is predominately involved in research in the area of embedded training. Previously, she worked for a defense contractor in training analysis and related work. She has a Master of Science degree in Industrial Psychology from Springfield College, and a Bachelor of Science degree in Psychology from the University of Florida.

THE DEVELOPMENT OF THE EMBEDDED TRAINING DECISION-AIDING AND RECOMMENDATION TOOL (ET DART)

P. Kelly Watson, Ph.D.,
Douglas Miannay,
Douglas Brashear
Lockheed Aeronautical Systems Co.
Marietta, Georgia

Richard A. Copeland, Jr.
STRICOM
Orlando, Florida

Nina E. Chatham
Orlando, Florida

INTRODUCTION

Embedded training (ET) research and development has been an important focus of military training for much of the decade of the 80's and into the present. The essentials of defining and implementing ET have been presented in publications such as the Army Research Institute's 10 volume series on Implementing Embedded Training (e.g., Finley et al., Strasel et al. 1988), as well as other work (e.g., Hoskin et al., 1989). Most recently, Witmer and Knerr (1991a) have produced a set of guidelines for making decisions about ET early in the weapon system acquisition process, unlike much of the previous work that tended to focus on the definition of specific ET characteristics and capabilities. What makes the decision guidelines produced by Witmer and Knerr so valuable is that they allow training planners, analysts and developers the opportunity to design-in ET from the earliest stages of system development.

These ET guidelines provide a set of questions that are associated with policy requirements as well as more traditional ET considerations, such as cost (see Witmer and Knerr, 1991b for a full description of the guidelines and their development). By separating the guideline questions into phases that are related to acquisition milestones, the authors have created a set of procedures that can provide an iterative description of the ET requirement for any given weapon system in the current or future Army inventory . With each successive phase or milestone, the additional information provides clarification and additional detail to this description.

The ET decision guidelines are a paper-based set of procedures that require the user to manually track the results (i.e., the recommended media, the ET alternatives, and any suggestions regarding training system requirements). Although the guidelines are not difficult to use as configured, there are additional capabilities that could be added that would expand the efficiency and usefulness of the process, such as sorting and summary routines for reporting. In addition, the process could be made much more efficient through automation. Figure 1 shows a sample decision flow from the ET guidelines. It is apparent from the examination of the guideline procedures that they lend themselves easily to automation.

BACKGROUND

At the Naval Training Systems Center (NTSC), Chatham (1992) developed an automated version of the guidelines as part of an overall multimedia production related to the topic of ET. Using a Windows-based software program called Toolbook, she created a prototype of a program that covered such topics as "Guidance" (containing reference documents), "Actual Systems" (using ET systems), "Research Topics", and "Considerations" (issues to consider when planning for ET). The "Media Selection" topic contained the automated guidelines. This program offered a Windows interface that allowed users to browse through the information alternatives before beginning the media selection process. The media selection itself was the direct incorporation of the Witmer and Knerr guidelines that are found in the original documentation. However, the program contained no capability for creating reports on the recommendations and results of the guided decision process.

Phase II, Block 3. Are other training alternatives supportable in terms of MPT and training facility requirements?

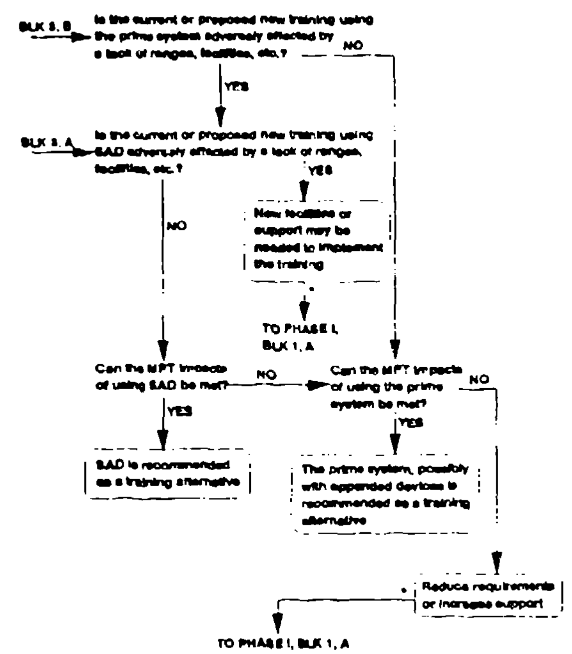


Figure 1: Embedded training guidelines (Witmer and Knerr, 1991a)

This was due to the nature of the program being primarily a demonstration of the kinds of PC applications that could be developed on this and other topics of interest to NTSC personnel.

Work at STRICOM was undertaken by Copeland (1992), and involved the assignment of the guideline decision rules to a forward chaining expert system shell called CLIPS (C Language Integrated Production System). An expert system developed using CLIPS consists of three components: a fact list, a knowledge base, and an inference engine. An individual fact from the ET guidelines fact list might consist of the decision tree node name, verbiage in the node (i.e., a question), type of node, and pointers to the next node (i.e., yes or no path). An individual fact contains all the specifics regarding each node. The STRICOM effort was developed with this in mind so that nodes could be easily added, changed, or deleted without changing the executable CLIPS code. The knowledge base consists of the rules required to execute the ET decision guidelines. A typical knowledge base rule would be the decision node rule which contains the specific executable CLIPS code called when executing a binary (yes/no) decision node. The inference engine is contained in the CLIPS shell and it controls the overall execution and forward chaining reasoning (see Figure 2). For a more detailed explanation of CLIPS, refer to Giarratano & Riley (1989).

The first step of the STRICOM effort consisted of a logic-only knowledge base, which contained basic decision tree logic and very primitive input/output, but which lacked any decision documentation. Also included in this step was the fact list, limited to the facts contained in Phase I of the ET guidelines. The second step of the STRICOM effort consisted of all decision tree logic, a textual input/output capability, and a decision documentation capability. Additionally, all facts associated with Phases I through IV of the ET guidelines were included in the fact list. This second step was completed by three U.S. Military Academy cadets during a summer internship at STRICOM.

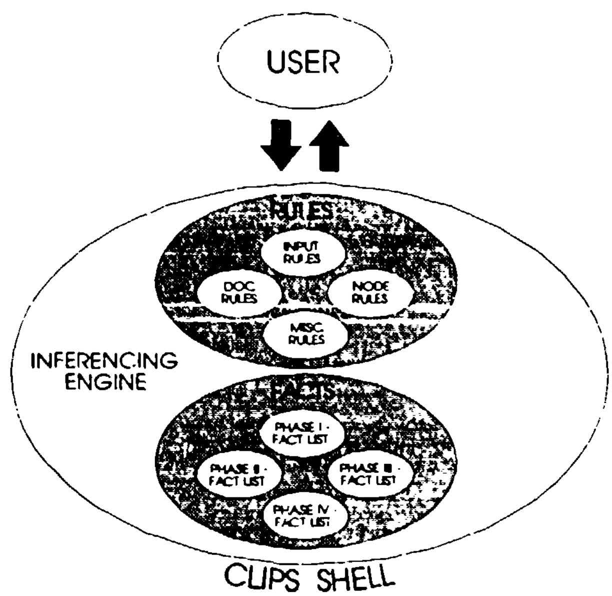


Figure 2: STRICOM program for automated ET guidelines.

The value of the STRICOM effort as an initial step toward automating the guidelines was twofold. First, it demonstrated the usefulness of grouping the fact list (nodes, node names, pointers) into one group, thereby allowing changes to be made without the necessity of changing the code. Second, it provided an area for documenting each decision or answer to the questions contained in the ET guideline documentation. These two features were perceived by the Lockheed developers to be valuable enough to be replicated in the new program that would be created.

The automation of the guidelines has been addressed by the two programs described above. However, the constraints on the resources available to both developers made the scope of the programs' effectiveness

somewhat limited. Creating the decision rules is the most straightforward aspect of ET guideline development; beyond that, however, are the issues of how best to create a program that provides more extensive assistance to training system planners. The Embedded Training Decision Aiding and Recommendation Tool (ET DART) is an attempt to do just that.

ET DART DEVELOPMENT

The ET DART was the first automated tool selected for development as part of the Lockheed Aeronautical Systems Co. (LASC) Independent Research and Development (IRAD) program entitled "Total Training Systems Integration" (TTSI). The intent of the TTSI is to identify which automated tools are needed to make the training development process more efficient, and which tools contribute to making the training system more effective. The current and future needs for training systems will be addressed by the development of additional tools and programs that will: 1) aid in the development and evaluation of training programs, and 2) provide expertise and support to training developers in making decisions that affect the training development process. Since the majority of the logic and underlying principles of the ET guidelines had already been developed by Witmer and Knerr, the remaining work was to focus on what additional capabilities, if any, should be developed as the guidelines were being automated. Figure 3 shows the design elements for the ET DART that were identified early in the planning stages of the program. The inputs for the program are identified in the existing ET guideline documentation, so the remainder of the design elements were the processes that the program would employ and the products users could get as a result. The ET DART development process was divided into three phases, which roughly corresponded to what we considered to be the most likely groups of work that could be accomplished during the year, given any additional work that had to be performed concurrently.

Phase I of the ET DART development process is intended to cover the development and incorporation of:

- the ET decision rules,
- a graphics user interface,
- a context-sensitive help function,
- a means for listing and saving user rationales for each decision made, and
- a Training Alternatives Summary Database, in which the recommendations made throughout each phase and block of the decision process are summarized for review.

Phase II of the program will focus on the development and incorporation of:

- on-line documentation access,
- scenarios for saving data and reports,
- decision rules for selecting media based on costs, and
- a process for making the final decision or selection of the type of ET system or training device to be developed.

The last period of development, Phase III, will focus on the development and refinement of:

- help information for users that includes both content assistance (how to use the guidelines) and program assistance (how to use the software), i.e., an electronic performance support system (EPSS)

Inclusion of Previous Work

At the beginning of Phase I, the programs by Chatham (1992) and Copeland (1992) were reviewed to determine what portions, if any, should be included in the ET DART. It was apparent from these reviews that each version had attributes that should be included in the ET DART, by either direct inclusion of the code or by use of the general principles. The decision concerning the development software to be used (see below), however, dictated that the best features of each predecessor version of the automated ET guidelines be included only in principle. The features to be included were the **graphic user interface (GUI)** from the Chatham program and the **decision documentation** from the Copeland program.

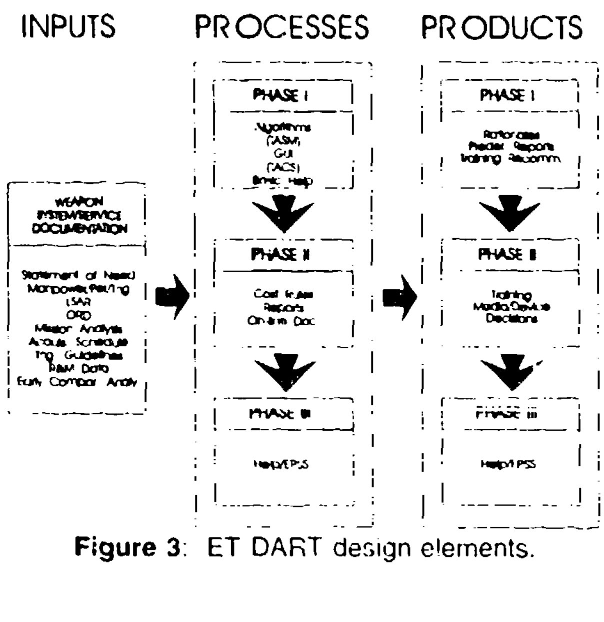


Figure 3: ET DART design elements.

The decision to incorporate these two features was very easy to make. The benefits of a GUI as the means by which users interact with the system are overwhelmingly obvious: the graphic interface provides color, form and an organizational structure that helps users learn about and use the system. In the era of Windows and graphics-intensive programs, these features are the standard means of providing the user interface. The decision documentation feature was a "smart" addition to the program from an administrative and management viewpoint, because it allowed users to provide justification for each decision. This feature allows reviewers of the process an insight into why the decision was made. By making the completion of the documentation for each answer optional, however, we can allow users to easily by-pass the documentation process if desired.

Terminology

As we reviewed the Witmer and Knerr guidelines, we noted that despite the overall applicability of the guidelines to other DoD settings, there were still "Army-isms": terms that were specific to the U.S. Army, such as "soldier", or references to the Army's maintenance practices. There were two ways of dealing with the changes to the terminology: first, we could make the terminology generic across the spectrum of DoD users, or second, we could change the terminology to suit the specific characteristics of each service, and have the service and its associated terminology selectable from the beginning of the program. The decision was made to make the terminology generic in order to avoid the complexity that was expected to occur by creating specific terminology (and possible additional guideline questions) for each service.

Weapon System and Service Documentation

The inputs to the ET guidelines consist primarily of weapon system documentation and policy and doctrine information that governs the specific service user, such as the Army. The ET DART development staff believed that efficiency could be increased if this information could be made available on-line to users. Users could then look up specific weapon system or policy information in order to make or support a decision. Although this capability is possible using programs in a DOS environment, Windows not only allows this feature, but is actually built around it. This Windows capability for allowing quick, on-line access to support information played a major role in the decision of the operating system environment to use for ET DART (see below).

Guideline Questions/Decision Rules

Of all of the aspects that were anticipated by the development staff to be relatively easy, it was the incorporation of the decision rules, i.e., the questions (see Figure 1). However, the rules were designed initially to be used as a paper-based procedure, and were laid out with that method in mind. We quickly found that there was more to the incorporation of the rules than merely typing them in as shown in the Witmer and Knerr document. For example, en route messages that are provided as the decision path is followed in the paper-based version must also appear in a results or summary area if they are to be of value to the user. In other cases, notes and responses marked with an asterisk (suggesting possible cost increases) that are produced in the process of using the guidelines must appear to the user of a computer-based version of the guidelines if they are to be useful. Therefore, the incorporation of the decision rules required the program developers to review the rules and the results to ensure that they would still be complete, meaningful and useful when transferred to the computer database.

Software Development Issues

At the outset of ET DART development, it was necessary to resolve certain software development issues, such as:

- Whether the application would be in DOS or Windows
- What software development tool(s) should be used
- What minimum hardware requirements would be imposed on the user
- How to produce a stand-alone, executable application
- Whether to consider the inclusion of multi-user or network capabilities and/or client-server architecture
- How to work within finite available resources for the development of the application

By analyzing the issue of available resources, it was possible to limit the other issues and choices significantly. Our available programming resources dictated that a software development tool utilizing the Xbase language would be necessary. This narrowed the possible selections, but still allowed several options for user interface:

Option 1. A text-based (DOS) interface utilizing Clipper 5.01 and Nantucket Tools II for programming the application.

Option 2. An intermediate step between a text-based interface and a GUI using Foxpro ver. 2.5 for DOS. (This interface provides pull-down menus, pop-up windows, point and click, click and drag, and other Windows GUI-like features, but runs in DOS and does not require the additional system overhead requirements of Windows.)

Option 3. The newly released Foxpro for Windows, ver. 2.5, providing an Xbase development platform while producing a true GUI and, with the additional Distribution Kit, a distributable .EXE file.

Thus, the first major issue became whether to develop the ET DART as a DOS-based tool or as a Windows application. There are several good reasons for each position: DOS can provide a "universal" operating system for the program, and can provide windows-like capabilities; DOS also may provide a faster program operating speed, although this is open to debate. Windows, on the other hand, appears to be the environment of the future, and provides a very attractive and capable means for housing the ET DART. Because Windows seems to be the environment of choice for most users and developers (or soon will be), the decision was made to develop the ET DART as a Windows application.

The next question to be decided was the development software to be used for the ET DART. The software selected for use on the ET DART was Foxpro for Windows; however, due to the need for expanded memory to accommodate the full capability of the software, the programmer staff had to use the DOS version of Clipper initially (Option #1 above), which provided usable code until the memory upgrade was accomplished. To make the ET DART an executable file, the Distribution Kit for Foxpro for Windows was obtained and used to produce the final ET DART products.

Although hardware requirements were somewhat a limiting factor for the development of ET DART, the project staff had to consider what hardware capabilities were likely to be available for ET DART users. For example, since Windows was the development and utilization environment selected, there was "automatically" a requirement for having at least a 386 processor with six megabytes of random access memory (RAM). However, the 486 processor is or will soon be the standard for most users, and greater numbers of Windows programs have mandated even more RAM as requirements for intended users. The decision was therefore pushed toward the higher end of the PC spectrum, as far as both processor speed and RAM requirements.

This is a supportable position as far as high technology organizations and users are concerned: organizations such as ARI and STRICOM often have higher end equipment to use in their jobs. However, users in the line organizations are often not so blessed. Their systems may be lagging somewhat in speed, memory, storage capability, as well as usable software. For this reason, the ET DART staff has reserved the option to create a DOS-based version of the ET DART, using Foxpro ver. 2.5, to ensure that users in the field will have access to a version of ET DART that offers basic decision-aiding capabilities.

In the interest of providing the most efficient, and ultimately, the most useful on-line help system, ET DART developers are incorporating an interim context-sensitive help facility into the ET DART tool. This type of help system will provide users with assistance in the use of incorporated application functions, not the "how to's" associated with embedded training analysis methodologies. An Electronic Performance Support System (EPSS), to be incorporated during a later phase of ET DART development, will provide user assistance with the analysis methodologies, plus all the functions supported by this interim help capability.

At completion of Phase I, the ET DART will offer a standard interface in which the user selects the appropriate "hot key" and is then presented with a window containing help suggestions for the function currently being used. Buttons appearing in the window will allow the user to select "QUIT" or "HELP INDEX". The HELP INDEX feature will present the user with an alphabetized listing of all help topics with standard "point and click" selection capability. If more than one screen is required for the help information, a standard scroll bar will appear on the right side of the window. When the help function is closed, the users are returned to the exact location from where they evoked the help. It must be stressed that this help capability is an interim solution and will be replaced with a more comprehensive EPSS system. In the meantime, ET DART users will have on-line help at their fingertips whenever they need application assistance.

Current Screen Design

The current screen design for the ET DART is represented in Figures 4, 5 and 6, which show:

- a screen showing the training element being evaluated (i.e., mission, function, or task), guideline question, answer, and rationale statement;
- a screen showing the results of a Block and Phase

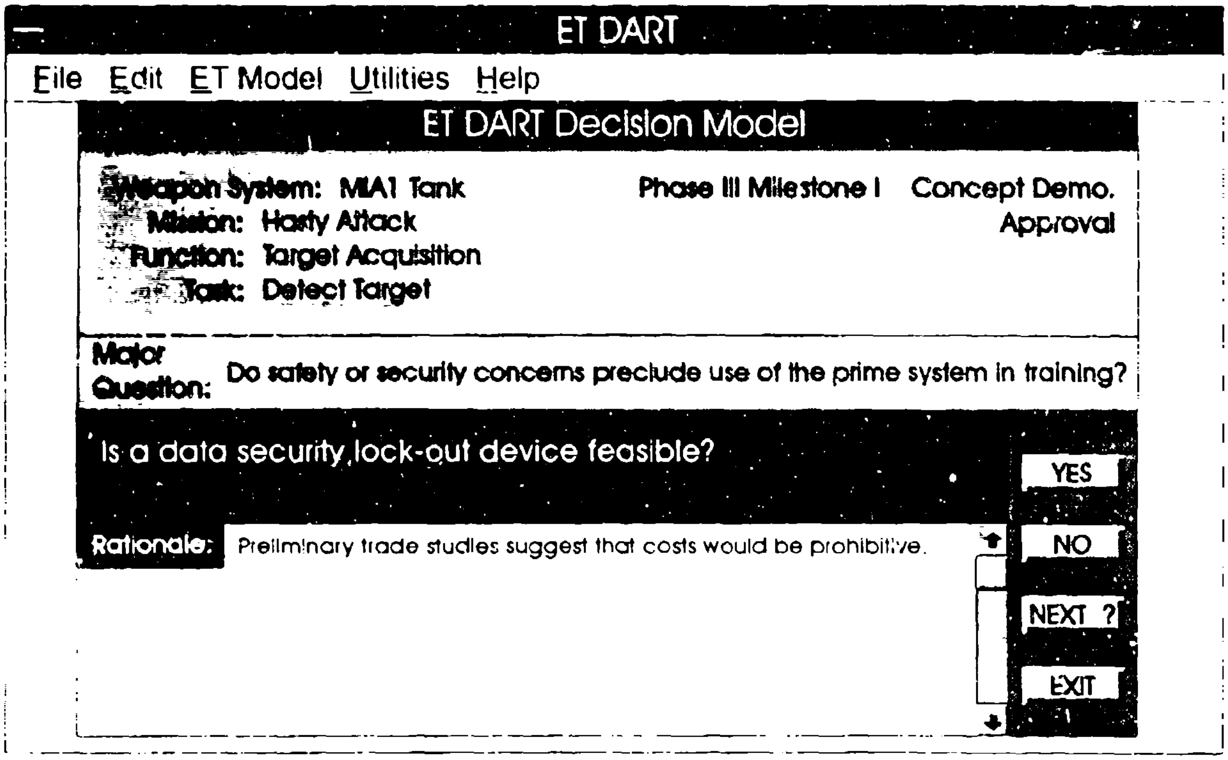


Figure 4: ET DART model guidelines screen.

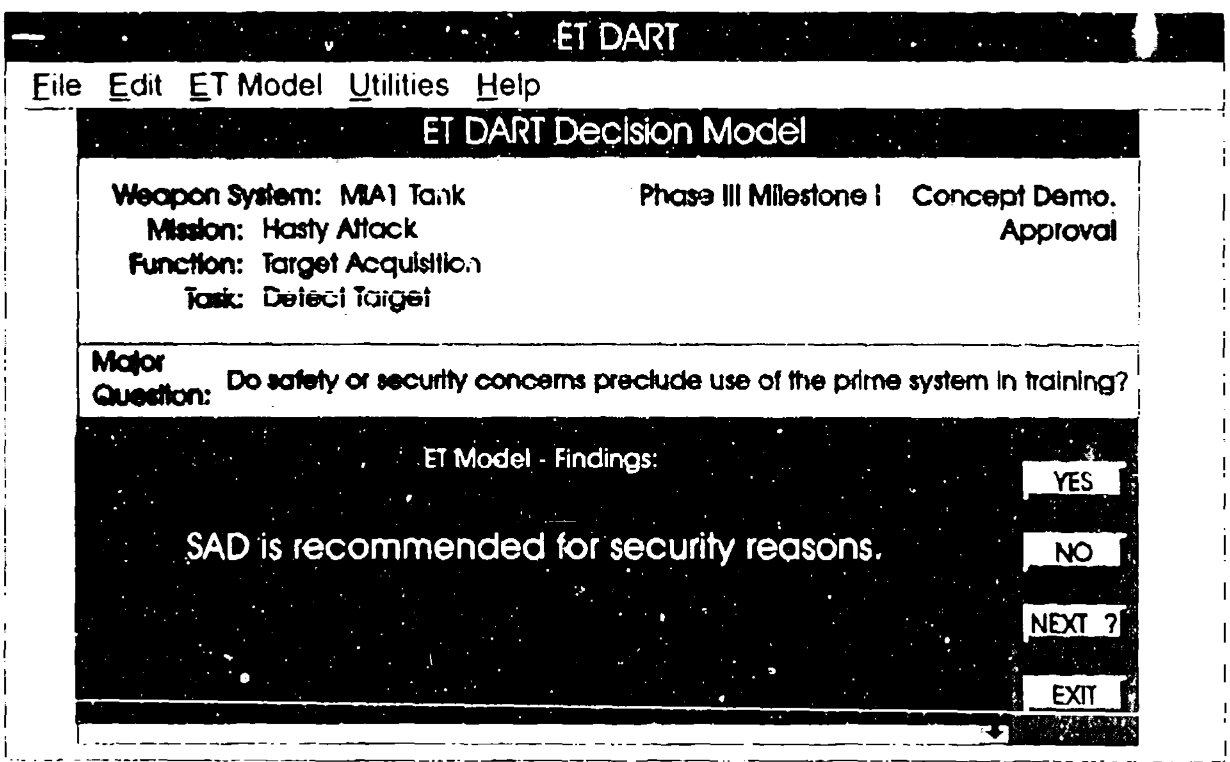


Figure 5: ET DART results screen.

11/30/93

ET DART
Embedded Training Analysis Report

Phase III Milestone I   Concept Demo

Weapon System: M1A1
Mission: Hasty Attack
Function: Target Acquisition
Task: Detect Target

Major Question:
Do safety or security concerns preclude use of the prime system in training?

Question:	Response:	Rationale for Response:
1. Is there a need to use simulation to reduce safety incidents or accidents?	YES	Probably only severity of accidents can be reduced.

Question:	Response:	Rationale for Response:
4. Is a data security lockout device feasible?	NO	Preliminary trade studies indicate cost would be prohibitive.

CONCLUSION

SAD is recommended for security reasons.

Figure 6: ET DART Block/Phase Report.

analysis;
- and the type of summary that results from the use of the guidelines.

These figures represent the ET DART as it is at this writing, and may evolve over the next few months. Additionally, the figures represent only the level of work that is or will be accomplished during Phase I of the program.

Figure 4 shows the screen used when the appropriate decision guidelines for the designated phase are provided for evaluation of the mission, function or task that has been input. Answers to the questions are provided by the user, along with any rationale that he may wish to provide. Figure 5 shows the results of the analysis. Figure 6 shows a report for a particular mission, function and/or task, and any additional messages encountered along the way.

FUTURE WORK ON ET DART

At this writing, the Phase II development goals will be focused primarily around the determination of the training devices/systems to be developed for the weapon system, using projected costs of the alternatives as the basis for the decision.

The last set of procedures in the Witmer and Knerr guidelines is the estimation of the costs associated with each of the recommended training alternatives identified in earlier steps. The cost estimates reflect non-recurring and recurring costs: design, development and procurement are non-recurring, and operation and maintenance are recurring. However, as Witmer and Knerr have noted, the interpretation of the cost data is the most difficult problem to resolve. The problem occurs when alternatives yield different levels of effectiveness, thus requiring different numbers of devices per alternative. Under these circumstances, the problem is one of cost effectiveness rather than just cost (Witmer and Knerr, 1991a).

The Phase II development of the ET DART should focus on the development of cost decision rules that provide an estimation of cost effectiveness. There are ongoing studies of cost and training effectiveness being conducted at the Institute for Simulation and Training at the University of Central Florida, under the sponsorship of Office of the Assistant Secretary of Defense (Force

Management and Personnel). This effort is directed at the development of a set of cost and training effectiveness methods and standards. In addition, the Aircrew Training Research Division of the USAF's Armstrong Laboratory is studying the cost effectiveness of different types of simulators and training devices in order to determine a standard set of methods for determining cost effectiveness of training. The ET DART project staff will incorporate as much of the results of these studies and other cost information as possible into a set of cost decision guidelines for use in making final training media decisions.

In the longer term (Phase III), the ET DART development staff intends to create and install an enhanced performance support system (EPSS) to replace the interim context-sensitive help system previously incorporated. The EPSS is a step forward in on-line user assistance, an initiative that addresses not only what a user needs to know, but how and at what rate they need to learn. In effect, the EPSS for the ET DART is a learning tool.

The ET DART is, naturally, centered around the determination of embedded training system requirements, and should be used early in the weapon system acquisition process. However, the recommendations and other resulting information that come from the use of the ET DART guidelines include stand alone devices, actual equipment training, and other alternatives as well as embedded training recommendations. In this regard, the ET DART can provide results that are applicable to other media selection processes and programs. However, it should be noted that since most media selection programs are task and/or objective driven (i.e., the medium is determined by the task or objective properties, such as type of learning, etc.) rather than policy, implementation, availability, etc., the results of the ET DART would probably be used mainly either as validation of other media selection program decisions, or as inputs to the media selection program for determination of the specific media type to be used.

SUMMARY AND LESSONS LEARNED

As part of the Total Training Systems Integration IR&D program, the ET DART development staff at Lockheed is creating an automated version of the embedded training decision guidelines developed by Witmer and Knerr (1991a). The development of the ET DART program, based in part upon earlier work by Copeland (1992) and Chatham (1992), will be performed in three phases, each of which will build upon and enhance earlier program capabilities. In addition to the incorporation of the decision algorithms, the ET DART will include a means for documenting each of the decisions made in answering the guideline questions, and a report production capability (Phase I). Later in the development cycle, the program will include the cost effectiveness rules or guidelines that will assist users in making final media decisions, and will contain the capability for using on-line documentation, i.e., the weapon system and service inputs (Phase II). The final phase of development will focus on the inclusion of an EPSS for providing users with both program-related and content-related help.

Finally, in the process of developing the program, we encountered several "learning experiences" that we felt should be passed along.

Lesson # 1. Do the basic version first, then enhance the program if you can, or if it's practical. When the project staff began the design of the program, there was a tendency to want to jump from basic coding to the development of major enhancements. Experience to this point suggests that when the baseline code is completed, it is easier to plan tasks that accurately reflect the scope of effort needed to change the program. For this reason, it was decided to produce the program in phases.

Lesson #2. There is a big difference between paper-based processes and products and computer-based processes and products (and getting from one to another). The nature of computer programs, and our reasons for using them, should suggest that we might not want to simply translate paper-based procedures to a computerized version of the same thing. Our realization of this principle came as we reviewed the decision rules, and the Training Alternatives Summary Matrix of the original Witmer and Knerr guidelines. The matrix is a good representation of the summary of results of the decision-making process, but only for the paper-based version of the guidelines. Computers can present the same information in a variety of user-selected formats, and can present additional information as required. It therefore made no sense, as we discovered, to try to merely replicate the matrix when we could design a much more efficient and effective method for presenting the data.

Lesson #3. Don't assume users know everything about computer programs, or care. As we developed the program, the user interface became an important issue. As a staff who ranged from knowledgeable programmers to mere computer users, we often found ourselves designing and developing a program that assumed too much about the user's familiarity with

Windows or computer programs in general. We therefore had to back off occasionally from the program to evaluate the user friendliness of the interface, and change it, in some cases, to ensure the usability of the program.

ACKNOWLEDGEMENTS

The authors would like to thank Cathy Howard for her help with the preparation of this paper, particularly the graphics, and Soo Lee Davis, Jeff Mazikowski, and John Sarabia for their assistance in development of the CLIPS based ET guidelines.

REFERENCES

Chatham, N.E. (1992). Embedded training. [Computer program]. Orlando, FL: Naval Training Systems Center.

Copeland, R.A., Jr. (1992). Automated ET guidelines. [Computer program]. Orlando, FL: STRICOM.

Finley, D.L., Alderman, I.N., Peckham, D.S., and Strasel, H.C. (1988). Implementing embedded training (ET): Volume 1 of 10: Overview. (Research Product 88-12). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Giarratano, J. & Riley, G. (1989). Expert systems: Principles and programming. Boston, MA: PWS-Kent Publishing Company.

Hoskin, B.J., Jorgensen, W.F., Manglass, D.A., Reynolds, R.E. (1989). Lessons learned from currently fielded Navy embedded training systems (Technical Report 89-011). Orlando, FL: Naval Training Systems Center.

Strasel, H.C., Dyer, F.N., Roth, J.T., Alderman, I.N., & Finley, D.L. (1988). Implementing embedded training (ET): Volume 2 of 10: Embedded training as a system alternative (Research Product 88-22). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Witmer, B.G. & Knerr, B.W. (1991a). A guide for early embedded training decisions. (Research Product 91-14). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Witmer, B.G. & Knerr, B.W. (1991b). Guidelines for embedded training decisions. Proceedings of the 13th Interservice/Industry Training Systems Conference. (pp.63-70). Orlando, Florida. American Defense Preparedness Association.

Dismounted Infantry: Indispensable to the Virtual Battlefield

Elton C. O'Byrne

Marine Corps Base, Camp Lejeune, North Carolina

Infantry on the Virtual Battlefield

Abstract

Today's infantry is more important to battlefield success than at any time since 1865. Forces are more widely dispersed because of the lethality of modern weapons. Dispersed, taking advantage of micro-terrain, able to maintain their mobility in any climatic context, dismounted infantry brings intelligence, flexibility, and killing weapons to the critical place at the appropriate time to define the battlefield and gain the victory. No model that denies dismounted infantry their proper place on the virtual battlefield can be validated. This paper defines the simulation requirements for infantry participation on the virtual battlefield of the advanced distributed simulation (ADS) environment. Current simulators pertinent to the this environment are identified, and placed in a conceptual framework that will enable dismounted infantry to participate accurately on the virtual battlefield. A phased implementation plan highlights the upgrade path from currently available trainers to the complete integration of foot mobile infantry in all phases of the ADS environment.

Dismounted Infantry: Indispensable to the Virtual Battlefield

Dismounted infantry fights on foot, using only the terrain, natural or modified, for cover and concealment. They may have arrived at the battlefield by foot march, or might have been transported by helicopter, truck, infantry fighting vehicle, tank or any other mechanical means, but they fight afoot. Dismounted infantry fights with hand weapons, rifles, hand grenades, machine guns, mortars, anti-tank and anti-aircraft rockets. While they will request and coordinate, by radio and telephone, support from other combat arms, they will also fight with knives, rocks and fists. Modern dismounted infantry can provide, at any spot on the battlefield, weapons capable of defeating any element of a combined arms army.

Satisfaction of a legitimate, joint force mission will be implemented in terms of possession of certain terrain or cultural features. Possession, that is control, of a feature is demonstrated by placing upon that feature an infantry unit. If that unit can be sustained on that feature, the objective is secured and the mission accomplished. This was the heart of the dilemma faced by General Schwarzkopf at the end of Desert Storm. Although VII Armored Corps units had reported the area "secured", they did not control it. The lonely Iraqi commander sitting on the airfield did. When the cease fire was announced with this individual still controlling the airfield, General Schwarzkopf was in a very uncomfortable position. (Schwartzkopf, 1992, p474ff) Had the Iraqi commander been of sterner stuff that feeling would have been made even more severe.

The fully implemented ADS environment requires that all forces be manned, so that such factors can be more accurately modeled. The simulation of air combat has been adequately accomplished for several years. Ships and tanks are being modeled quite well. Dismounted infantry has been defined as "too hard to do". The very elements, however, that make it difficult are the very ones that make the infantry man so difficult to defeat, and thus indispensable to the virtual battlefield.

Simulation of dismounted infantry is more difficult than larger entities such as ships, planes, or tanks. The infantry world requires extremely subtle graphics and portrayal of non-regular movements. Infantrymen are not very big and each one has to be killed individually. This requires that a relatively small projectile be delivered into a critical portion of his anatomy. The pertinent visual environment is composed of a nearly infinite number of elements. Differentiation between them is based on extraordinarily subtle cues, such as a different shade of green on leaves being used as camouflage, or the disturbed dirt where a mine was dug in. The infantry men disperse widely and spend most of their time prone, or otherwise behind the terrain. They are dirty and therefore present

very little contrast with the terrain. This all requires an inordinate degree of simulation sophistication.

The movement of aircraft and other vehicles are controlled by a seated operator, manipulating Cartesian oriented controls. Conversely, infantrymen slither into an infinite variety of positions so as to gain as much protection as possible while firing their weapons. Their weapons are at best fixed on a single point of support. Most are in essence, free floating in the environment. They are aimed by direct connection between the eyeball and a sight, thus no electronic presentation is available from which to gain X,Y data.

As a result of these difficulties, most current infantry simulators use canned scenarios stored on laser disks. This provides the requisite visual fidelity, albeit at the cost of firer-target interactivity. Rather than model the pressure inputs used to modify the weapon aim, the most accurate systems project a beam from the muzzle of the weapon onto a reactive surface. The combination of these processes provides a relatively accurate critique of the firing solution. Return fire can be modeled in the same manner. The tradeoff point for interactivity versus visual fidelity is un-specifiable because it varies with the exercise objectives. The advantages of unit interactivity in the Advanced Distributed Simulation environment should make current technology acceptable for exercises above the individual level.

Battlefield mobility is not solely the speed of the entity. A MIG-25 Foxbat traverses the battlefield at 1,600 MPH, yet has little immediate impact. As General Schwarzkopf found to his chagrin, the ability to drive over the battlefield at thirty or forty miles per hour doesn't allow the commander to label a road junction or airfield as controlled. Mobility is measured in the ability of a force or element thereof, to move combat power to an advantageous position. Thus an A-10 Wart Hog, capable of less than 400 MPH has more tactical mobility than the Foxbat, because it can directly and quickly, deliver firepower at entities in important tactical positions.

Mobility is more than a rate. It must also be extended through time. A mechanical vehicle such as a tank or aircraft must return to a relatively secure area to conduct maintenance and refuel. The infantryman can stay on the scene of the fight. With ammunition, and water resupplied, an infantry company will stay the night and probably the next day. Given more water and a bit of food, they will stay forever.

Military units are defined by their primary weapon. The relative importance of each weapon has varied over the years. Man first fought on foot with swords, knives, and spears. The invention of the stirrup made the cavalry supreme. Gunpowder provided a means to send projectiles against the cavalry. The size of the first guns required the organization of artillery units to move and fire them.

These units were dominant on the battlefield until improved gunpowder and metallurgy allowed development of infantry rifles that could outrange them. Pickett's charge on the last day of Gettysburg marked not only the high water mark of the Confederate States of America, but also the high water mark of the artillery until wireless communications allowed forward controlled indirect fire.

Infantry, using man portable small arms, then held the balance of battlefield power until the internal combustion engine and caterpillar tracks made possible large armored fighting vehicles. These tanks were armored to defeat machine guns and because of the tracks were able to maneuver off roads. In the same period, automatic weapons and aerial bombs were added to aircraft. To late to affect the operations in World War I, great promise was foreseen for aerial delivered ordnance. During the period between the World Wars, military theorists promised that the tank and airplane would achieve dominance over the battlefield. Their ability to maneuver and fire was to make the battlefield a vast sea over which the victorious armored and aerial forces would reign supreme.

Although such mobile operations were important during the early stages of World War II, as Paddy Griffith has said, "Contrary to the common impression that Second World War battles were easy, f-st-moving and decisive affairs we find that they were in reality protracted, grueling, nerve racking and costly." (Griffith, 1990 page 118) That is the infantry presence was dominant. The record is replete with recent examples of infantry, fighting on their feet, but holding their own against all measure of forces. In the Western Desert, Rommel was twice lured into attacking a well defended position. First, at Tobruk in April 1941, and then at Alam Halfa in '43 his mobile forces were forced into attacking intelligently developed, infantry based defenses. In both cases his panzers were defeated, resulting in the first case in channelization, and in the second in defeat. (Liddell Hart, 1970 page 296, Wellington,1993, page50ff, Mellenthin, 1971, page 170ff).

Always underlying their armoured thrusts, the Russian Army especially in the defensive used infantry intelligently. Such a defense was the key factor at Kursk in July 1943. Major General von Mellenthin, Chief of the General Staff, 48th Panzer Corps:

> "The Russian High Command had conducted the Battle of Kursk with great skill, yielding ground adroitly and taking the sting out of our offensive with an intricate system of minefields and anti-tank defenses." (Mellenthin, 1971, page 277)

By the summer of 1944, the rocket based infantry anti-tank weapon had appeared. It was a significant element in the German defense against Operation Market Garden. General S. L. A. Marshall, the pre-eminent historian of twentieth century infantry combat, identified the panzerfaust as one of the keys to

the defeat of operation Market Garden. By delaying for twenty hours, the initial movement of the armored forces toward the paratroopers at Arnheim, dismounted infantry armed with Panzerfausts (Marshall, 1963 page 9) ensured that Arnheim would be a "Bridge Too Far".

After World War II, technical development added a command link to the anti-tank rocket. This extended the infantry's tank killing range to the limit of visibility. These weapons were tactically significant in the Egyptian success during the 1973 Yom Kippur War. Their plan called for dismounted infantry to cross the Suez Canal and form tight bridgeheads. ALthough contrary to Russian doctrine, they planned to defeat the initial Israeli armored counter-attack with infantry fired anti-tank rockets, thus avoiding the mobile armored combat which the Israelis desired. The plan worked, defeating the Israeli counter attack and holding the bridgeheads until Syrian failures in the north lead to the defeat of the Arab force. Nowhere during the Yom Kippur War was an attack against a well organized defensive system successful. ..."the attack was always destroyed in a conclusive manner".(Griffith, 1990 page 173)

The development of radio communication and indirect fire methods returned the artillery to a larger role for nations that could afford them. Forward observers now became the only artillery men required to actually see the enemy. Provided with a radio, a map, and defensive protection, the forward observer could direct artillery fire in devastating effect on enemy forces. This provided to the dismounted infantry unit an additional mission. During the Vietnam war, then Lieutenant General Ray Davis, Commanding General First Marine Expeditionary Force, used infantry to establish observation posts in enemy held areas from which forward observers, defended by infantry, could control indirect fire upon enemy forces. In modern maneuver warfare, such positions will be held by companies or even battalions, depending upon the terrain to be held. Inserted by helicopter or other mechanical means, the infantry will defend the observation post so that the artillery car kill the enemy and direct b[illegible]ttlefield intelligence can be gained.

In addition to killing the enemy with direct and indirect fire, infantry units are the major providers of tactical intelligence. Despite the need for remotely sensed indirectly gathered intelligence, the most valuable source of information is the infantryman actually looking at the battlefield. Well planned, aggressive patrolling, when combined with accurate reporting provides operational planners with the information needed to multiply the effect of the forces' weapons. The tactical problems encountered by American forces in Korea and Vietnam were largely the result of enemy control of the night. Without the ability to gather intelligence by patrols, commanders were forced to rely on indirect sensing of the battlefield This usually left them a cycle behind.

While offensive operations are necessary to win a war, defensive operations must be an integral portion of the commander's plan. Defense is more efficient in it's use of combat power. Weapons are most effective when sited to cover an expanse of terrain over which the attack must come. The terrain can be modified to facilitate the application of fire. Units in a defensive posture can be more closely supported by air and indirect fire weapons than can moving forces. For these reasons, the modern battle is most often decided by placing infantry on key terrain, without possession of which, the enemy cannot achieve their objectives.

Proponents will claim that the defensive has no place in maneuver warfare. In most cases however, it is possible to use maneuver to gain advantageous position, often considerably to the rear of the enemy, then meet his advance with effective defense. A defensive fulcrum or hinge must be provided to anchor the enemy force to allow deep turning movements that are a significant part of maneuver warfare. Thus Lee had to establish a strong defensive position before Jackson could make the deep turning movement at Second Manassas. General Boomer had to make a limited attack to make possible the wide movements of the XVIII Airborne Corps and VII Armored Corps in Operation Desert Storm.

Such weapons and mobility make today's infantry more important to battlefield success than at any time since 1865. Forces are more widely dispersed because of the lethality of modern weapons. Dispersed, taking advantage of micro-terrain, able to maintain their mobility in any climatic context, dismounted infantry brings intelligence, flexibility, and killing weapons to the critical place at the appropriate time to define the battlefield and gain the victory. No model that denies dismounted infantry their proper place on the virtual battlefield can be validated.

Valid simulation of modern warfare requires that all forces on the real battlefield be so modeled that their tactical presence interacts with other forces in the same manner as actual combat and the virtual battle therefore yields results highly similar to combat. A simulation exercise without dismounted infantry cannot be validated as a model of real world combat. Accurate definition of the battlefield and realistic actions from air and mechanized forces cannot be achieved without dismounted infantry. Current visual systems and terrain models can't completely support the individual dismounted infantry and the micro-terrain that define small unit infantry operations. It is however, possible to simulate a squad in a defensive position.

Reduction in battlefield troop density been a permanent feature of modern warfare.[figure 1] The increasing lethality of weapons has made dispersion necessary. Whereas the Napoleonic army required three shoulder to shoulder ranks to maintain continuous fire, modern weap-

ons enable a four man fire team to develop continuous killing fire over a 45 degree sector.

Modern communications allow small units to disperse and still maintain contact with supporting agencies and higher headquarters. Just as important, modern training produces motivated, intelligent infantrymen capable of operating in small groups. The apparently overly rigid chain of command noted by outsiders in good infantry units is the result of the process of accountability needed to develop small unit leaders capable of such command. The increased lethality of fire and the improved communications demand and allow small units to have unprecedented importance on the modern battlefield. As a result, modeling the battlefield does not require battalions assaulting on line. It can legitimately be accomplished with currently available squad level trainers linked to the virtual battlefield via the Distributed Interactive Simulation (DIS) standards.

Simulation of modern warfare requires accuracy in the depiction of terrain, of weapons, entity movement, target presentation, and environment. While all vehicles require such accuracy, the infantry requires that the DIS exercise be accurate to different measures.

Terrain's effect on combat comes from it's effect on maneuver and as cover. Such effect is the result of the size and location of terrain features. This effect can be measured by the model proposed by Simpkin, that is, by bandwidth and roughness, so that terrain's effect of forces is highly similar to that experienced in the real world. The first parameter, wavelength, represents the average distance between two similar peaks. The second parameter is roughness. This is the amplitude, or height, of the terrain profile with respect to each waveband. Such description of terrain affords a more discriminating tool than does the more usual Defense Mapping Agency description in terms of scale and contour interval. The measure of terrain bandwidth and roughness will not yield the sine wave patterns of music or electricity. (Simpkin, 1985 page 58ff) Because terrain is the result of many disparate dynamics, the resultant of our pseudo mathematical analysis can be more accurately described by the theory of chaos. The generation of terrain measured in bandwidth and roughness and expressed by fractals should accurately represent terrain for any tactical entity.

Weapons need to be modeled for their effective range, lethality, and rate of fire. Anti-armor weapons must be modeled for lethality vis a vis the target vehicle and angle of hit. The obscuration of the sighting mechanism by environmental concerns is an important consideration. These are all rather accurately described in ballistic and other systemic material and need not further discussion herein.

Entity movement must be appropriate to inherent capability as modified by the specific terrain and environ-

ment being encountered. All entities on the electronic battlefield must be in jeopardy of being killed or wounded by enemy fire. This should reflect individual entity protective measures and exposure to fire. Local meteorological phenomena e.g., rain, fog, and temperature and their effect on the terrain must be modeled to a high degree of fidelity.

Additional conditions must be measured to ensure fidelity. Killing fires must be credited. These might be direct fire weapons fired at visible enemy, or final protective fires directed on an azimuth per the unit fire plan. Credit must be provided the defending force for slowing the assaulting forces by appropriately located field fortifications such as barbed wire, abatis or ditches. Minefields, indirect, and suppressive fires are just as important as aimed direct fires and must be credited for their impact, not only in casualties, but also in modification of the opposing force movement and observation capabilities.

Each side's fire must be related to the opponent's. This includes shots passing close enough to an individual to dissuade him from increasing his exposure enough to fire more accurately. Especially critical is measurement of fire superiority that could prevent movement by the inferior side to either deny or facilitate the final assault. Fire must be constrained by appropriate logistics support factors. Unit of fire and basic loads prescribed in operations orders must be depleted by actual shot count and augmented by planned or ad hoc resupply. This might well be constructive, as the staff work and supply movement can modeled quite accurately.

Given this conceptual framework, the specific needs for simulation of dismounted infantry in defensive combat can be stated.

Terrain- Maximum requirement: Dynamic with bandwidth and roughness of one meter required for ultimate simulation. Minimum requirement- Static terrain with bandwidth and roughness of one hundred meters. This low fidelity terrain will increase the effectiveness of the defense by reducing cover and concealment usable by individuals. Weapons to be modeled and available to the infantry are listed in Figure 2.

Entity movement must be modeled to a high level of fidelity, both as to speed and limitations of terrain and environment. Full simulation fidelity would be represent individual movement. A satisfactory minimal system would restrict individuals to a single defensive location, but would allow hemispheric firing and observation.

This analysis leads to a phased introduction of dismounted infantry. to wit:

Phase 1. Restricted defensive combat-Integrate current trainers into a single facility per the DIS protocols to allow normal fire control by squad leaders. Return fires to be scored by indirect means. Terrain bandwidth and

roughness of one hundred meters acceptable.

Phase II. Mobile defensive combat-Full perimeter position, company commander able to organize defensive position to match orders, projected terrain, and visible enemy situation. Leader to be able to move fires as well as troops within the perimeter. (Requires dedicated facility with 10,000 square feet of floor space and a domed roof.) Return fires to be scored by indirect means. Terrain bandwidth and roughness of one hundred meters acceptable.

C. Offensive combat-Virtual reality, with forces allowed full movement upon the virtual battlefield, full-dynamic terrain with one meter bandwidth and roughness.

While the final phase is not possible with current resources, it is possible to accomplish the interim phases. History has proven that the dismounted infantryman is indispensable on the modern battlefield. Technology cannot currently fully simulate their actions. Current systems can however, simulate company level strongpoint defensive operations. This partial solution will provide a legitimate model of modern warfare until full virtual reality is possible.

Dismounted infantry fight in the most difficult environment for simulation. Dynamic terrain, individual movement, weapons control subject to three degrees of freedom make demands upon the visual system and computational power. But these are the same elements that make the presence of dismounted infantry indispensable on the virtual battlefield. "...the foot infantryman with his "computer-brain" has proved a tougher species than Fuller ever imagined him." (English, 1981 page 201) Leaving him off the virtual battlefield is unthinkable.

References

English, John A. "On Infantry" New York:Praeger 1981

Griffith, Paddy. "Forward Into Battle".Novato,CA:Presidio.1990

Liddell Hart, B.H. "History of the Second World War". New York: G.P.Putnam's Sons 1970

Marshall, S.L.A. "Battle at Best".New York: William Morrow.1963

Mellenthin, Major General F. W. von. "Panzer Battles" A Study of the Employment of Armor in the Second World War".New York:Ballantine Books. 1971

Schwartzkopf, General H. Norman. "It Doesn't Take a Hero"New York:Bantam.1992

Simpkin, Richard E. "Race to the Swift:Thoughts on Twenty-First Century Warfare".London: Brassey's Defence Publisher. 1985

Wellington, Mike. "Sorties from Tobruk" Military History Magazine January 1993.

Figure 1
Troop density per mile of battle front

Year	Troops/Mile
1800	20,000
1870	12,000
1917	2,500
1980	1,000

English, 1981, page XIX

Figure 2
Infantry Weapons

Weapons organic to the U.S. Marine Corps rifle company
- M16A2 Service Rifle, 5.56mm
- M203 Grenade launcher, 40mm
- M249 Squad automatic weapon, 5.56mm
- M60 Medium machine gun, 7.62mm
- SMAW
- AT-4
- Heavy machine gun M2 .50 and MK19 40mm
- 60mm mortar
- Dragon ATGM

Weapons liable to be attached to the company.
- TOW ATGM
- Stinger Anti-air missile

Weapons fired from outside in support of the company.
- 81mm mortar
- Artillery of all calibers
- Naval gunfire
- Close air support- Bombs; napalm; rockets; 20mm, 30mm, and [illegible] cannon strafing;

SARA CAR DRIVING SIMULATOR: AN AMBITIOUS RESEARCH AND DEVELOPMENT TOOL

Michel LACROIX, Jean-Pierre GAUBERT
THOMSON-CSF, Département Simulateurs
Cergy, FRANCE

Pierre GAURIAT
Groupement d'Intérêt Economique SARA
Arcueil, FRANCE

ABSTRACT

This paper describes the objectives and main features of the French car driving simulator for research applications, called SARA project. Requested by the French Transport and Safety Research Institute (INRETS) and the two French car manufacturers (PEUGEOT S.A and RENAULT) this simulator allows:

-Safe and accurate evaluation of driver's attitude in various situations,
-Accurate traffic engineering research,
-Engineering evaluation of vehicle design.

This simulator is a technological state of the art design, as far as it incorporates :

-A specific motion system including a 6 DOF motion platform on top of a large X-Y linear displacement system and a specific vibration device,

-A wide field of view visual system including a 180° front field of view and rear view mirror scene displays. This system is based on a Computer Image Generator and a collimated display system,

-A specific software and database development center allowing the preparation of real time experiments and analysis of their results.

ABOUT THE AUTHORS

Michel LACROIX is the Project Engineer of the SARA car driving simulator at THOMSON-CSF, Département Simulateurs, France. Since joining THOMSON-CSF in 1973, he as been responsible for the development of control loading systems, motion systems and display systems. He holds an engineering degree from the Institut National des Sciences Appliquées de Lyon (FRANCE).

Jean-Pierre GAUBERT is the Program Manager of the SARA car driving simulator at THOMSON-CSF, Départememt Simulateurs, FRANCE. Previously, he worked with the VISA Computed Image Generator R & D team. He holds a diploma in Electronic Engineering from the Institut National des Sciences Appliquées de Lyon (FRANCE).

Pierre GAURIAT is Research Director at the Computer Research Center of the French Transport and Safety Institut (INRETS, Institut National de REcherche sur les Transports et leur Sécurité). He is currently the SARA Project Manager. He received his diploma from the French Aeronautics High School in 1969. In his previous activity, he has been involved in computer graphics.

SARA CAR DRIVING SIMULATOR: AN AMBITIOUS RESEARCH AND DEVELOPMENT TOOL

Michel LACROIX, Jean-Pierre GAUBERT
THOMSON-CSF, Département Simulateurs
Cergy, FRANCE

Pierre GAURIAT
Groupement d'Intérêt Economique SARA
Arcueil, FRANCE

INTRODUCTION

Operator-in-the-loop simulation has been widely used in research and development in aeronautics, as well as for training in a number of highly technical fields. Along with recent developments in computer science (CPU performance and the generation of synthetic images), it has provided the automotive industry with such tools as, for example, the simulators of VTI in Sweden, and Daimler-Benz in Germany.

In '87, French Government Agencies and car manufacturers decided to combine their efforts into a national project for a car driving simulator. Technical feasibility studies were undertaken in '87 and '88; in '89 the French Transport and Safety Research Institute (INRETS) and the two French car manufacturers PEUGEOT S.A. and RENAULT started cooperation to attain the financial and material capabilities needed to conduct such a high technology project, and to bring together work forces from different origins and of multiple experiences in a common development team.

The Simulator Division of THOMSON-CSF was chosen as the contractor for the design and manufacture of the simulator. The three partners, INRETS, PSA and RENAULT, agreed to undertake several specific parts, among which the real-time vehicle dynamic model.

GOALS AND STAKES

The purpose of the SARA Project is to build an advanced car driving simulator sophisticated enough to be used as a research tool on driver behaviour by laboratories and government agencies, and also as an engineering tool by car manufacturers during the design of a new car. This equipment should help gather skills, and should act as a mainspring for research in the field of vehicle development as well as road safety.

Since car safety is a major concern for French researchers and car manufacturers, the design and operation of a full-task simulator is a real need.

Challenges

Design of the car driving simulator will require specific research and development requiring various disciplinary approaches: mechanical engineering, mathematical models, ergonomics, acoustics, applied experimental psychology, computer science, etc.

It should be noted that the complexity of dynamic simulation of ground vehicle motion is far greater than those required to represent aircraft motion.

It has become apparent that car driving simulators have higher demands in general performance terms than aircraft simulators, for example response time is shorter and the simulation cycle time will be typically of the order of ten milliseconds.

The scientific goals are threefold:

1) Simulator Design: the vehicle application can be addressed by modern techniques, but it will be necessary to improve the latter in certain specific areas (acceleration and visual cue generation),

2) Integration Of Knowledge Of The Industrial And Scientific Partners: this is not a minor

challenge, and it also imposes upon the partners the problem of confidentiality of their own results. Consensus must be reached on the modularity of the model as a whole, and on some specific modules, such as tyre road contact. It will illustrate French know-how on the subject, while every partner will retain the possibility of using its own solutions,

3) Simulation With Respect To Driver Behaviour: the studies necessary to validate the experiments undertaken on the simulator will rather provide fundamental study of human behaviour than a mere statistical description.

For the automotive industry, the goals are twofold:

1) Simulator Design: a whole set of techniques, methods or knowledge will have to be collected and formalised. In addition to being necessary for the design of the facility, they will prove useful beyond this field. PSA and RENAULT will undertake mathematical models for vehicle dynamic behaviour in cooperation with one another and with a tyre specialist,

2) Simulator Use: it will be a sophisticated laboratory associated with road trials; it will reduce the costs and the duration of development, a noticeable improvement for the future evolution of vehicles in European research programmes and industrial projects.

Examples

At the international level and in the field of automotive and driver studies, we can mention two designs which seem to be good examples of such car driving simulators.

The VTI car driving simulator (Vag och Trafic Institut) with a three degree of freedom motion base: the main criticism which can be levelled at this good simulator is its inability to depict a very complex road scene. In other respects the simulation, although limited, is of perfectly adequate quality for the studies involved. In 1990 and 1991 this success led the VTI to design and build a new simulator based on the same principles.

The Daimler-Benz simulator (DB research center in Berlin) better known than the above: the general architecture is comparable to aeronautic simulators with a 6 DOF motion system. The design is both remarkable and impressive. However, the quality of operating and image simulation systems reveals the improvements that could be made in the ability of the motion base to reproduce certain violent transient phenomena which occur when driving at limits or in an emergency situation.

Research Areas

Three research fields are concerned:

- Vehicle,
- Road environment,
- Driver behaviour in a realistic driving situation.

The simulator is a research tool well adapted to a "system" approach. It will facilitate program and research themes combining the above three fields.

1) Vehicle Design - This important goal has enabled definition of the basic level of simulator performance (motion system, mathematical model validity and the quality of visual restitution).

The simulator will be used as a test laboratory in order to complement the traditional tools of the manufacturers. Its main use, however, will be the development of new architectures and new solutions. The final adjustment of the vehicle which requires a very subtle appreciation of car behaviour and consequently accurate modelling of both car and tyre effects is not envisaged at the moment.

The simulator will host the following studies, which need both a complex tool and the "decision" of a human driver:

When in the first step of new car conception, the comparison of various technological solutions, and the validation of technical choices,

- At road behaviour study level, a "credible" numerical model for professional testers will help develop parametric studies. It will also help study the evolution of parameters inaccessible when in real situations,

- For the design of new vehicle architectures involving active subsystems (e.g. active suspension), the simulator will be used, for example, for the analysis of the consequences of the architecture on the primary safety of the vehicle. Qualification studies will be undertaken on the simulator, with, according to the user's wishes, the mathematical model of the subsystem or the actual subsystem,

- Closely related to the latter, the study of modern instrument panels, although possible with part-task simulators, will nonetheless use the advanced simulator when it becomes necessary to assess the impact of new driving aids on driving behaviour.

2) Research On Driving Tasks And Active Safety- Without a full-task simulator, several studies are too complex or dangerous to be conducted on the road. For example, an integrated tool with the possibility of simulating a realistic driving situation is necessary for research on driving behaviour:

- Emergency driving manoeuvres,

- New electronic driving aids and the capacity to minimize the effects of loss of concentration,

- Effects of tiredness, alcohol and drugs on the driver,

- Typology of behaviour and driving strategies, impact on energy consumption in given situations,

- Detection of objective indications and of parameters constituting a feeling of safety.

3) Road Engineering For Road Safety Administration And Research Laboratories - The advanced car driving simulator can be used to conduct road design and road sign legibility research, at three levels :

- Fundamental research, related to the aforementioned studies, for which the driver's sensations are critical,

- Applied research: the simulator will increasingly be used for studies on visibility and marking, especially in borderline situations (night, atmospheric disturbance, etc.), with the purpose of designing new tools and improving placement methodologies,

- "In the field" studies: important road programmes will need "in situ" studies on the simulator for specific problems. For example, the modification of geometrical parameters will be undertaken after the study of the behaviour of several drivers on several vehicles (light vehicle, trucks, convoys, etc.).

Obviously standard road or highway projects do not need this advanced facility. But design of critical sites and improvement of important black spots will be demonstrated and tested on simulator before building, with positive effects on time and cost.

Let us point out the advantages of the car driving simulator as opposed to more traditional investigation tools, in particular experimentation on track or road:

- The situation studied can be reproduced,

- An experimental scenario can be reproduced with possible variations,

- Tests can be conducted on a model of the vehicle without having to actually build it,

- A degree of freedom exists in the modification of those variables which cannot be easily managed during actual tests, and parametrical analysis is easier,

- It is possible to analise dangerous situations safely.

SIMULATOR DESIGN

The SARA car driving simulator is a technological state of the art design, as far as it incorporates:

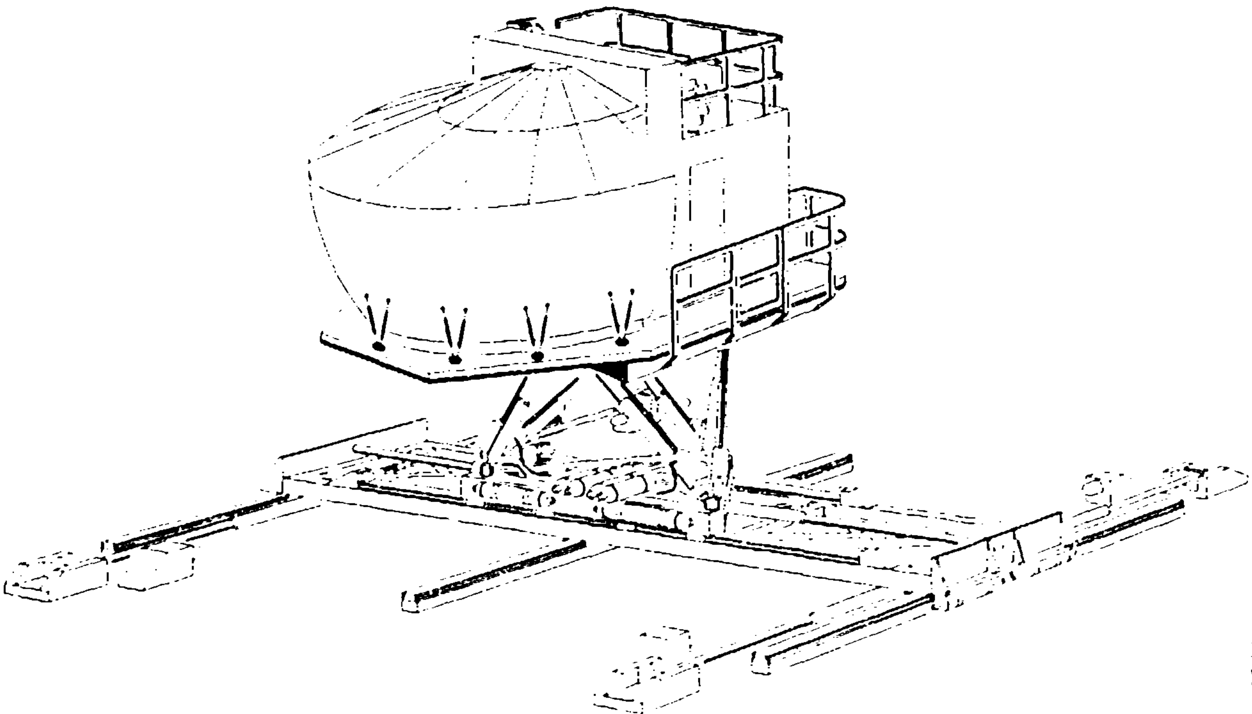


Figure 1 - Car Driving Simulator Motion System

- A specific motion system including a 6 DOF motion platform, a large X-Y linear displacement system and a vibration device.

- A wide field of view visual system including a 180° front field of view and rear view mirror scene displays. This system is based on a Computer Image Generator and a collimated display system.

- A specific software and database development center allowing the preparation of real time experiments and the analysis of their results.

MOTION SYSTEM

The motion system shall provide large amplitude and low frequency motions in lateral, longitudinal, pitch, roll and yaw as well as low amplitude and high frequency motions in the six degrees of freedom. This is achieved by the use of a 6 DOF synergistic motion system on top of a large X-Y motion system and a specific vibration device fitted under the car structure. A physical layout of the motion system is shown in Figure 1.

Performance Requirements

The motion system must generate a sufficient level of motion cueing fidelity to evoke natural driver response behaviour. According to the level of cueing fidelity sought, motion system performance requirements can be established.

The approach taken to establish the performance needed is to simulate the 6 DOF motion system on top of the large X-Y linear displacement system, and to drive it with accelerations that have been recorded in a real vehicle with linear and angular accelerometers placed under the driver's seat for several manoeuvres such as lane change, step steering, emergency braking. For the fidelity factor sought i.e. the ratio of recovered acceleration to drive acceleration, the required motion system excursion, velocity and acceleration envelopes are obtained.

Because the motion system has limited horizontal excursions, long-term lateral and longitudinal accelerations cannot be maintained using only linear motions and, cabin tilting must be added. Lateral or longitudinal motions reproduce with no lag the start of the

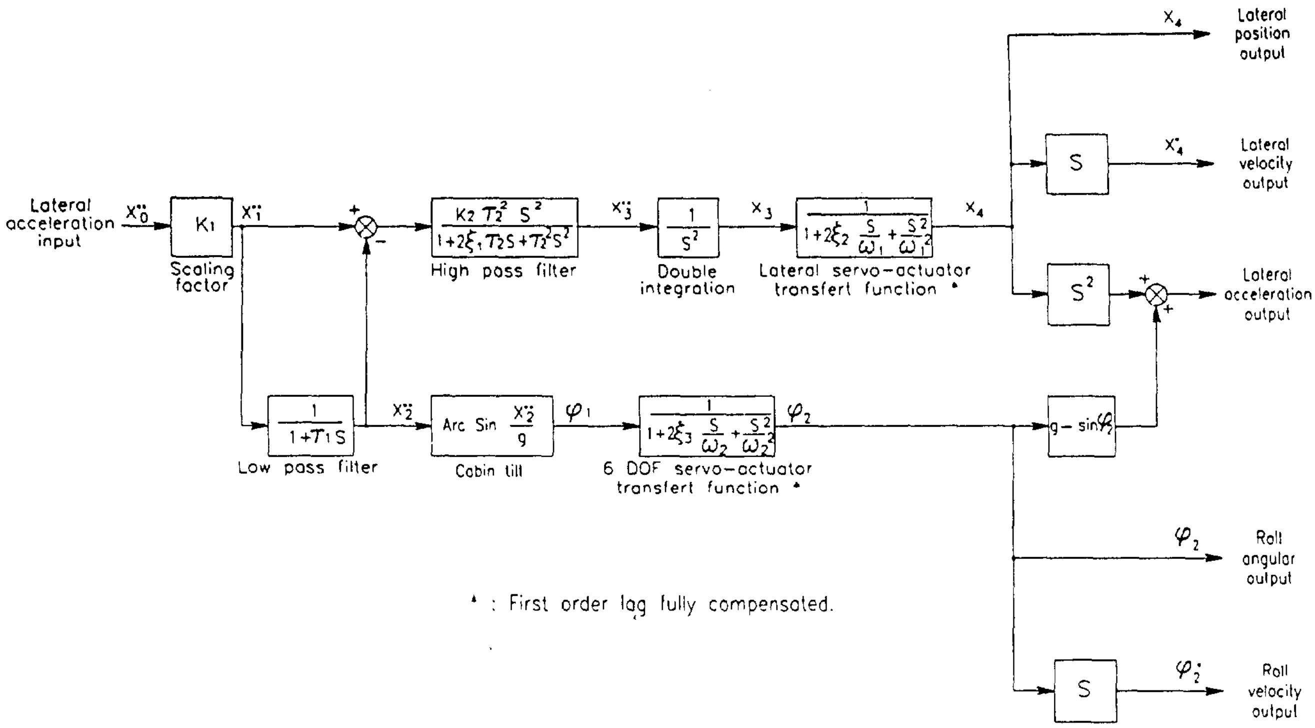


Figure 2 - Lateral Motion Block Diagram

acceleration which is maintained by roll or pitch cabin tilt.

The system modelling block diagram for the lateral motion is given in Figure 2. Low pass and high pass filters are used to separate long and short term accelerations(low and high frequency terms). The lateral acceleration input is first filtered with a first-order low pass filter to obtain the longterm acceleration for calculation of the roll tilt angle. After multiplication by the 6 DOF servo actuator transfer function, the roll tilt angle of the cabin is obtained and the resulting simulated lateral acceleration due to gravity is computed.

The filtered lateral acceleration used to compute the roll tilt angle is subtracted from the lateral acceleration input and the result is filtered with a second order high pass filter and integrated twice to get the lateral position of the cabin. After multiplication by the large X-Y motion system transfer function, the lateral position is obtained and differentiated twice to get the simulated lateral acceleration.

Simulated lateral accelerations from the large X-Y motion system and the 6 DOF motion system are summed and compared to the lateral acceleration input. Gain and time constants of high pass and low pass filters are adjusted to get a smooth acceleration transition between the large X-Y motion system and the 6 DOF motion system and to maintain the tilt rates below the human threshold of perception.

Using the simulation model, motion system excursion, velocity and acceleration envelopes have been computed for lane change, step steering and emergency braking. Results of the simulation for a lane change are given in Figure 3. This shows a fidelity factor of 0.7.

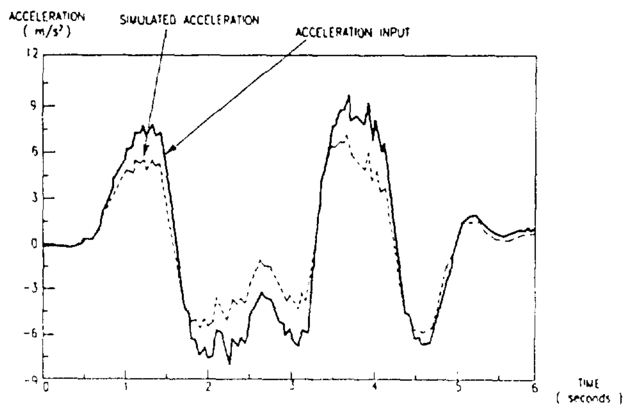


Figure 3 - Simulated Lane Change Response

Large X-Y Motion System

The main difficulty is to meet large excursions, high velocities and low acceleration

noise levels. The acceleration noise mainly comes from drive vibrations transmitted by the mechanical structure and turnaround bumps due to Coulomb friction and backlash in the drive and carriage systems. For the drive system, the use of hydraulic or electric motors with rack and pinions or wire cables has been eliminated since they lead to friction, backlash and vibration problems. The use of a direct linear acting device has been preferred with the choice of either a linear electric motor or a hydraulic actuator. The latter has been selected because it is a proven technology, used on 6 DOF motion systems for flight simulators.

The hydraulic actuators are double-acting equal area cylinders fitted with hydrostatic bearings for zero Coulomb friction. One actuator is used for the lateral motion drive and two identical actuators working in parallel are used for the longitudinal motion drive.

The guidance system consists of a lateral (Y) carriage and a longitudinal (X) carriage sliding on one another. In order to get a coulomb friction as low as possible, the X-Y motion system is guided by means of hydrostatic bearings.

The lateral carriage is equipped with three double-acting hydrostatic bearings that are fitted under the triangular base of the 6 DOF motion platform to support vertical compression and extension forces, and with two identical hydrostatic bearings that are fitted on one side of the base to support longitudinal forces.

The longitudinal carriage beam structure is equipped with 6 double-acting hydrostatic bearings to support vertical forces and 3 double-acting hydrostatic bearings to support lateral forces.

6 DOF Motion System

The 6 DOF motion system is identical to the system used on commercial aircraft flight simulators except for the geometry which has been modified to get ± 30 degrees usable angular excursion in pitch and roll. This system incorporates hydrostatic actuators, ultrasonic position transducers and a digital servocontrol system with force feedback.

Vibration Platform

The vibration platform has 6 degrees of freedom and it is used to reproduce vibrations from 3 Hz to 30 Hz. This platform is of the same design as those used on helicopter flight simulators and it can generate accelerations up to ± 1.5 g. The vibration platform is fitted inside the motion platform frame and it supports only the car structure. This solution avoids vibration of the display system at high frequency.

DISPLAY SYSTEM

The display system comprises a collimated wide angle display system for the view through the front windows of the car and a real image projection system for the rear view mirrors. The complete display system is contained in a light-tight enclosure and its general arrangement is shown in Figure 4.

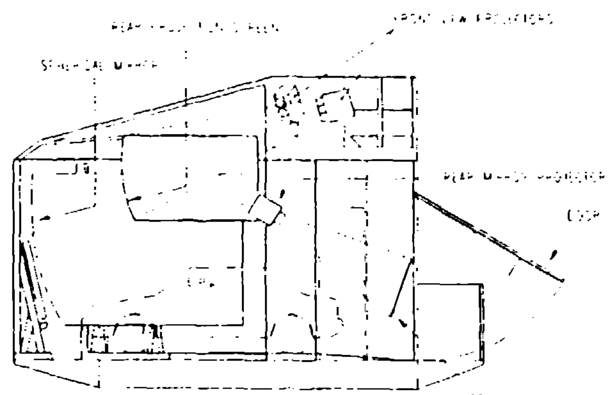


Figure 4 - Display System General Arrangement

Collimated Wide Angle Display System

The system gives a continuous panoramic field of view of 180° horizontal by 40° vertical and has a resolution better than 3.0 arc minutes.

The visual scene displayed is produced by three CGI channels. Each channel drives a high brightness CRT projector mounted above the car shell on a rigid gantry. The projectors produce images on a spherical projection screen fitted above the car's front windows. The images of each visual channel are carefully aligned and matched to produce a continuous scene on the projection screen which is viewed by the driver via a wrap-around spherical concave mirror.

The mirror is so positioned in relation to the screen that a collimated image can be seen by the driver. In comparison with a real image projected onto a dome screen, the collimated image recreates the time needed for the eyes to accommodate from an object at infinity to the dashboard, and it also gives the driver the feeling of being immersed in the visual scene.

Rear Mirrors Display System

This system comprises a flat screen located behind the car and two projectors mounted above the car structure and under the rigid gantry that supports the three projectors of the collimated display system. The two projectors correspond to left hand and center rear view mirrors. The projectors used are commercial video projectors since edge matching and high brightness are not needed.

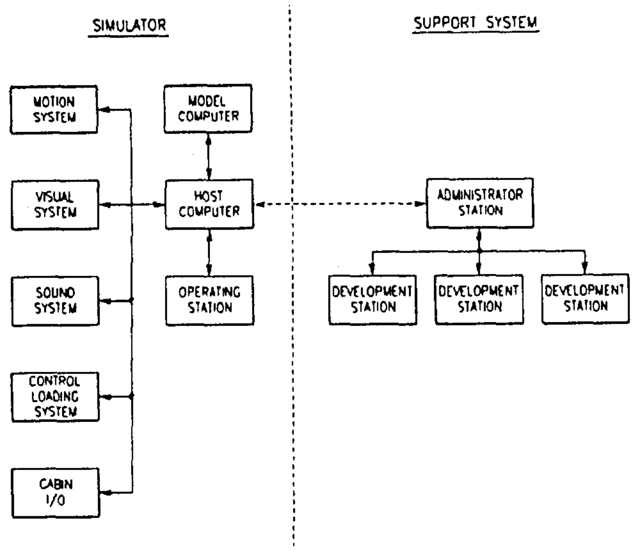


Figure 5- Computing Center Architecture

COMPUTING CENTER

The simulator is designed to perform studies on driver reactions, and on vehicle and road environment designs. These studies require multiple experiments that must be prepared, performed and analysed.

The preparation phase consists of the assembly of the different data necessary to define the experiment i.e. the vehicle dynamic model, the visual database, the scenario, and the informations to be recorded during the real-time execution of the experiment. Once the experiment has been performed on the real-time simulator, the recordings must be checked and analysed.

Because the preparation and analysis phases take a certain amount of time to be performed, and because three different users can be working simultaneously, the simulation center houses two different systems as shown in Figure 5 :

- The real-time simulator dedicated to the execution of experiments. This system can only be used by one user at a time,

- The support system dedicated to the preparation and analysis phases where the three different users can work simultaneously using a secure software environment.

Exchange of data between the two systems is operated through the control of the administrator by using a magneto-optical disc device.

Real-Time Computing Center

One of the most important features of the SARA simulator is to be a research simulator. This means that its structure must be sufficiently evolutive to allow future upgradings.

From this point of view the computing facility comprises a host computer dedicated to the standard simulation software (considered as the basic system software insofar as it should not support major evolutions) and an associated computer, called "model computer", housing the vehicle dynamic model software (considered as the applicative software insofar as it will support continuous improvements).

1) Host Computer. Its functions are:

- The real-time monitoring of the simulator. This function is dedicated to the management of the different software modules running on the computer, and to the correct synchronization of subsystems such as visual system, motion system, control loading system, model computer and operating station during the simulation

cycle. It also manages the input/output data flow between the host computer and the subsystems,

- The simulation of the environment that generates all the data required by each subsystem. It provides the visual system with the visibility conditions and the position of the observer and of the different moving objects. It also provides the sound system with the sound parameters corresponding to the environment and the driver's actions. One of the main simulation functions of the host computer lies in the motion system control software which takes into account a large range of parameters (driver's actions, vehicle characteristics, driving database informations), in order to compute those commands most adapted to reaching the actual motion cue. This control software can be adapted according to each manoeuver characteristic by simply modulating the control of each degree of freedom available on the motion system. The modulation algorithm is predefined by the user when preparing the experiment,

- The scenario management that manages the experiment conditions (vehicle paths, meteorological conditions, animations, data to be recorded) which have been set up during the experiment preparation. Note that experiment execution is controlled through the operating station where most of the information can be displayed.

2) Model Computer. This computer houses the dynamic vehicle model. It provides the host computer (through a reflective memory) with the parameters necessary to compute the complete environment and perception cues. A user-friendly Model Generator, allowing description and modelling of a vehicle with the precision needed by engineers whilst remaining acceptable for real time simulation, is used to generate the code that will be run by the model computer. This software will allow the description and the simulation of any kind of rigid or flexible constrained multibody system on which a set of complex subsystems acts. These subsystems will be real or modelled devices of any kind (mechanical, hydraulic, electric, electronic, etc.).

Support System

As discussed earlier this development facility is a multi-user device. For this reason, the system is based on a secure B1 environment. Three identical configurations are provided. They all include the software tools necessary to perform the development of the experiments and the analysis of the recordings. Main development tools are:

1) Database Creation Tool which allows the creation of the environment database used by different subsystems of the simulator. Starting from a unique model of the environment, this tool generates four distinct and coherent databases:

   - the visual database used · the image generator to compute the visual scene,

   - the sound database used by the sound system to generate the correlated noises,

   - the driving database used by the host computer to determine the data to be used by the vehicle dynamic model (charateristics of the terrain) and the motion system control software,

   - the operating station database used to display the map of the database (control of the experiment),

2) Scenario Creation/Modification Tool which defines all the events that will occur during the experiment, i.e. moving vehicle path, correlations between the moving object attitude and the environment (driver, database, etc.), animations (level crossing, lights, etc.), evolution of visibility conditions (fog, light sources, etc.), data to be recorded during the experiment, etc.,

3) Motion Control Software Tool which identifies the required algorithms to be used during the real-time experiment. It provides the user with all the information and parameters to be set up in the motion control software, in order to get the optimum response of the motion system. The adequate

motion algorithms are then pre-programmed and evaluated before the experiment is performed,

4) Recording Analysis Tool that analyses the data recorded on the support system after the experiment has been performed on the real-time simulator. The replay function makes it possible to repeat the experiment by using the parameters recorded during the real-time experiment (driver's commands for instance). It is then possible to analyse the results by means of general purpose tools making it possible to select and sort the required data.

CONCLUSION

Those designing and utilizing simulators must be constantly concerned about the possibility of transposing studies carried out on a simulator to the real world, validating models and justifying the compromises which must be made concerning simulation itself.

Applications of simulation to motor vehicles can in this respect benefit from the long experience which the aeronautical world has accumulated. This field is, however, extremely specialized and it appears difficult to transfer existing simulation techniques as they stand. Each design has an innovative function.

The SARA simulator confirms this rule and it is a high technical challenge. Most of its subsystems use the most advanced technology. The motion system and its control software, the visual and display system, the software architecture will make SARA a unique research facility.

SARA is also the opportunity to introduce in the scope of simulation some less demanding activities such as road engineering in order to validate the design of critical sites without heavy investments.

Motion simulation of other vehicles (for instance helicopters) should benefit from this unique motion system technology which will allow the improvement of motion cueing simulation.

Cognitive Fidelity in the Design of a Maintenance Troubleshooting Trainer

J.S. Bresee and W.W. Wagner
Delex Systems, Inc.
Vienna, VA

ABSTRACT

Previous papers have explored the concept of cognitive fidelity and its application to training in decision-making skills. These papers have described the concept of cognitive fidelity and its value in ensuring the realism of information as an essential factor in decision-making training. Devices designed for high cognitive fidelity would provide a user with highly realistic information, but might not require a physical environment of corresponding realism.

This paper reports on the design and development of a device for training the troubleshooting of an aircraft fuel system. The paper's initial focus is on the design choices made to ensure that cognitive fidelity remained high under conditions which sharply constrained physical fidelity. The paper shows how the functional requirements of specific training objectives were used as a basis for design specifications.

The development of the troubleshooting trainer is described, identifying the key design choices, and the way in which cognitive fidelity was used as the basis for selecting between specific design alternatives. Specific features examined include simulation of test procedures, simulation of related systems, and trainee interface.

ABOUT THE AUTHORS

Mr. Bresee is employed by Delex Systems, Inc. as a training analyst and designer specializing in training requirements analysis, training systems design, and training capability development for aviation systems, both military and commercial. He is presently concerned with supporting the FAA's model Advanced Qualification Program for airline cockpit crew training.

Ms. Wagner is a simulation engineer with extensive experience in real-time modeling. Working for Delex Systems, Inc, she has been developing aircraft system simulations and student interfaces for military aircraft part task trainers, and for commercial aviation maintenance training. She is the principal design engineer for the family of training products which includes the DC-8 fuel system simulator.

Cognitive Fidelity in the Design
of a
Maintenance Troubleshooting Trainer

J.S. Bresee and W.W. Wagner

INTRODUCTION

Training environments employing simulation traditionally have been made as realistic as technology would allow, with designers taking every opportunity to add any feature of the task environment which could be represented. The effectiveness of this approach is inarguable, and for jobs in which the consequences of a mistake are high, it clearly makes sense to allow students to master component skills in safe environments before applying those skills where their correct performance is critical. What often has been in question is the cost of realism, and how this realism should be measured.

This paper describes the design and development of a simulation-based training environment where training is administered over a wide geographic range, and where cost considerations demanded effective use of an installed hardware base. The nature of the task and the stringency of the constraints forced the design team to achieve a very clear focus on training goals and resulting design criteria. The resulting product provides an information-rich environment by means of fully functional simulation to achieve highly effective training at a controlled cost.

DIMENSIONS OF FIDELITY

The most common dimension for measuring "realism" is the accuracy with which the physical environment of the job is represented. The importance of physical fidelity grew from an "identical elements" theory of transfer that is nearly a century old (Thorndike, E.L. and Woodward, R.S., 1901). Adoption of this theory caused simulation and training engineers to focus on maximizing the number of characteristics of the training environment that match those of the job environment. This approach has been used extensively in simulation for flight crew training, and has been beneficial, proving to be both practical and effective as guidance for simulator design. However, the elements which have been made identical are nearly always defined in physical terms, leading to elaborate installations and mathematical modeling which extends beyond perceptible limits.

Fidelity, or the lack of it, has been under scrutiny in the technical literature for many years. In 1980, D.R. Farrow surveyed the then-available literature in a presentation to the Society for Applied Learning Technology (SALT) conference of that year. In this paper, he noted that some writers had begun to consider the reduction of fidelity along some dimensions when high realism in irrelevant areas might actually detract from learning.

The preeminence of physical fidelity was entirely appropriate as long as psychomotor skills were the major training requirement. It is still highly appropriate, in many cases. When the desired trained behavior is to coordinate the manipulation of flight controls with cues received from visual and instrument displays - in other words, basic airplane handling skills - it makes perfect sense to strive for a simulated environment where these cues are made to resemble the cues available in actual flight. Physical stimuli elicit the onset of controlling actions, and provide data for the evaluation and modification of the action or initiation of subsequent action. Manual control of the aircraft can be seen as a continuous series of approximations and corrections in response to physical feedback from the environment. The more accurately the physical environment is simulated, the more accurate the feedback, and hence the more representative the task performance.

The training of maintenance technicians focuses on an entirely different skill set. While psychomotor skills are important, the major training requirement is cognitive. Fault isolation, or troubleshooting, is a problem-solving task,

making use of highly detailed information on the state of the system being maintained. A prior paper made the point that problem-solving behavior improves as the range of information and techniques available to the technician increases through experience (Bresee and Greenlaw, 1992). Training provides synthetic experience, and the appropriate training environment increases its effectiveness.

While psychomotor training requires fidelity of physical stimuli, decision-making training requires fidelity of information. The term *cognitive fidelity* has come to be used for this dimension. Cognitive fidelity is taken to mean the realism of information content, presentation, and management options that is present in a simulated task environment. This topic was explored in a previous publication (Bresee, J. and Naber, M. 1991), where the training requirements of tactical decision making tasks were examined. Here, a decision was described as a choice to be made that is not dictated by a procedure. These relatively unconstrained choices are guided by heuristics which are formed from fundamental principles, as modified by the results of practice. For this practice to be effective, and for the resulting skills to reliably transfer to the job environment, cognitive fidelity must be high.

It has been argued that decision making, especially in team environments, is more or less effective to the extent that a common mental model of the task environment is shared by members of the team (Cannon-Bowers, Salas and Baker, 1991). This same point of view was advocated by Judith Orasanu at the International Airline Transport Association symposium on aircrew training in September of 1992. In her presentation, Orasanu advocated the building of shared mental models as the precursor for cockpit crew problem-solving tasks. She cited communication of accurate, realistic information about the problem situation as the critical element in building these shared mental models through which problem-solving heuristics were applied. Both researchers would seem to support an emphasis on high cognitive fidelity as a foundation for accurate and useful mental models.

When cognitive fidelity is maximized, it does not always follow that physical fidelity is also high. Part task trainers optimized for decision training have been successful in fulfilling their mission without high physical fidelity in every aspect of their design. This can be illustrated by examining the aspects of fidelity included in training devices which have been designed for training tactical skills. These are decision making skills, and are improved through practice in handling information of the nature and quality of that received in an operational setting. This has some similarity to maintenance troubleshooting behavior. In both cases, trainees must learn what information to select, as well as how to act upon it.

DESIGN EXAMPLE: A FUEL SYSTEM SIMULATOR FOR TROUBLESHOOTING TRAINING

The design process followed for an aircraft maintenance training system provides an example of how optimizing cognitive fidelity increases the effectiveness of decision-making training aids. In this case, the training requirement grew out of maintenance operations for an older aircraft, where the organization was losing troubleshooting expertise through retirement. Developed for the DC-8 fuel system, this trainer was designed to provide maintenance trainees with troubleshooting expertise through practice on simulated equipment. This trainer has also been put to use, providing training and job support for maintenance technicians for over a year. It is considered successful by its users, having proved its effectiveness in supporting classroom training, individualized practice, and support for actual flight line troubleshooting.

The training requirements driving the design of this device were clearly cognitive from the first. Providing accurate information in a realistic manner was accepted as a design goal from the first. However, operational constraints were also operating in that the user had a geographically wide-spread student population, and an installed base of hardware whose use was desirable. If possible, the trainer should function on a DOS-compatible computer with an 80286 CPU chip. This made it clear that extensive hardware simulation requirements would be difficult at best to implement. The discussions quickly focused on isolating the

fidelity requirements which could be relaxed, and those that must remain stringent.

It soon became clear that every design consideration could be subordinated to the providing of information for the troubleshooting process. During troubleshooting, information is gathered from instruments and other data-delivering devices, but also from the physical condition of the system itself. This is also impacted by student entry level. If a component or system is unfamiliar, its physical status will not be readily perceived, and this itself becomes a training requirement. Entry level emerged as a key factor in making specific trade-offs between physical and cognitive fidelity factors.

As design discussions continued, the team was able to abstract an engineering rule-of-thumb for design with respect to physical and cognitive fidelity requirements: The importance of specific physical fidelity is reduced when components and tasks are familiar. Once the form of a control or display has been learned through repeated use, unless its exact form and function is task-critical, it is no longer cognitively relevant (salient) and need not be represented with high physical fidelity. However, when items of equipment - or displays on familiar equipment - are unfamiliar, their exact form and function are cognitively new, and physical fidelity becomes important as a component of cognitive fidelity.

The team also considered the physical requirements of the information acquisition and management tasks that comprise DC-8 fuel system troubleshooting. These tasks are performed within the cockpit environment, using the fuel panel itself. The trainer must not require the student to operate this panel differently than that of the aircraft; otherwise, false diagnostic cues may be introduced. This line of thought gave rise to another cognitive design rule of thumb: Reductions in physical fidelity cannot add distracting difficulty to task performance. For example, choosing to represent the fuel quantity gauges as a CRT graphic cannot result in providing different information about the rate of tank filling than would be shown by the actual aircraft instrument.

This rule of thumb was found to have a corollary: Limitations on physical fidelity must not result in excessively modified task performance. For example, a CRT representation of the fuel control panel cannot compress or alter the spatial relationship of components to the extent that actual diagnostic procedures cannot be authentically performed.

Here again are the four rules of thumb for cognitive design that have been discussed to this point:

1. The importance of specific physical fidelity is reduced when components and tasks are familiar.

2. When items of equipment or information displays are unfamiliar or task-critical, physical fidelity becomes important as a component of cognitive fidelity.

3. Reductions in physical fidelity cannot add distracting difficulty to task performance.

4. Limitations on physical fidelity must not result in excessively modified task performance.

These four statements relate the need - or lack of need - for physical fidelity to cognitive training requirements. Further reflection upon the nature of the troubleshooting training requirement added two more candidate design rules:

5. Cognitive fidelity is increased when all information normally available (both necessary and extraneous) during actual operations is present in the training environment.

6. Cognitive fidelity is increased when all control options and actions (both relevant and irrelevant) that are available during actual operations are present for training.

As the trainer design process continued, these rules of thumb proved increasingly valid. For example, it was quickly established that all trainees were highly familiar with the cockpit layout of the aircraft, and with the major components of the fuel system. The four principles

addressing physical fidelity limitations were completely applicable. A two-dimensional graphic representation of the fuel control panel was adopted. The circuit breaker panel, of keen interest in troubleshooting tasks, was considered to be so familiar that only reports of its condition were required. The team elected to use a pop-up window showing the condition of any individual circuit breaker upon query.

Schematic-like maps were used for the large components (tanks, pumps, lines) for troubleshooting tasks. However, the customer informed the team that not all students understood the structure, function, operation and control of all pumps, valves and sensors in the system. Therefore, some preparatory modules were added where the structure and function of each class of pump, valve and sensor was graphically modeled. Here, two-dimensional representations were used as a compromise. Higher physical fidelity was desired, but not considered practical.

Frame-oriented computer-based training (CBT) is often used for improving the look and feel of aircraft maintenance training. The customer had originally considered this approach. However, the essential nature of the troubleshooting task required a significantly more realistic approach than the paginated treatment that CBT often applies to a complex process. Troubleshooting is, more than anything else, a classical problem solving task. The trainer design had to support the basic requirements for solving problems:

- Complete information regarding the problem, both relevant and irrelevant.
- Unconstrained choice of action within the domain of the problem
- Accurate and appropriate knowledge of results.

These requirements coincide completely with the fifth and sixth rules of thumb mentioned above. This forced the design team to use a system simulation. CBT developers often produce products which appear to be simulations in that trainees interact with representations of controls, and see changes in control position or system state. However, in many of these products, the student may not deviate from the desired "pathway" of control actions. These products are often called "path simulations" to distinguish them from fully modeled "freeplay" simulations. Full and complete information about any system fault does not come - in any cost effective manner - from a set of pre-programmed faults and fixed diagnostic paths. It was clear that only a fully functional system simulation providing a freeplay environment would provide the necessary cognitive fidelity for carrying out the troubleshooting task.

The resulting design was a system simulation written in C in an MS/DOS environment using an EGA graphics interface. This resulted in a readily transportable product useable on a diverse installed base of computational equipment. Moreover, this product was useful for more than only training tasks. Because this trainer incorporates a full freeplay model of the fuel system, technicians have used it as a flight line performance support system. By using the instructor mode to selectively fail system components, technicians can confirm or disprove a tentative diagnosis of an actual operational problem.

SUMMARY

This paper has offered some concrete guidelines for the use of cognitive design principles in the design of a maintenance training environment. It has described a trainer designed in this manner, whose employment as an interactive job aid as well as classroom training device shows that high training value can be attained through part task trainers with relatively low physical fidelity, provided that cognitive fidelity remains high.

REFERENCES

Bresee, J.S. and Greenlaw, D.R. (1992). Cost efficient maintenance training through cognitive fidelity, Proceedings of the 14th Interservice/Industry Training Systems and Education Conference, San Antonio, Texas.

Bresee, J. and N. bor, M. (1991). Tactics as decision making: Issues in tactical training development, Proceedings of the 13th

Interservice/Industry Training Systems Conference, Orlando, FL.

Cannon-Bowers, J.A, Salas, E. and Baker, C.V. (1991) Do you see what I see? Instructional strategies for tactical decision making teams, Proceedings of the 13th Interservice/Industry Training Systems Conference, Orlando, FL.

Farrow, D.R. (1980). Front-end analysis of existing systems: research and development requirements. Proceedings of the Third International Learning Technology Congress and Exposition. Warrenton, VA. Society for Applied Learning Technology, 28-38

Orasanu, Judith. (1992) Crew problem solving and decision making, Presentation to the IATA Symposium on Crew Training, Stockholm, Sweden, September 30, 1992.

Thorndike, E.L. and Woodward R.S. (1901). The influence of improvement in one mental function upon the efficiency of other functions. Psychological Review, 1901, 8, 247-61, 384-95, 553-64.

DESKTOP SIMULATION FOR AVIONICS MAINTENANCE TRAINING

Scott D. Royse
Kevin J. Miller
Charles J. Zinsmeyer
Southwest Research Institute
San Antonio, Texas

ABSTRACT

The F-16 C/D Avionics Intermediate Shop Maintenance Procedures Trainer (AIS-MPT) represents a significant advance in the application of desktop simulation techniques to a training task that has previously been addressed only through interactive computer-based training (CBT). In order to meet requirements for a high-fidelity simulation of the AIS computer system using captured operational data, the simulator was designed using a digital multimedia representation of the four automatic test equipment (ATE) station types that is controlled by a simulation execution environment written in Ada. The result is a unique combination of real-time simulation programs and multimedia-based "simware" running on a networked, dual-CPU student station, and providing a true training simulator for the AIS and for F-16 line replaceable units (LRU).

The AIS-MPT provides the training environment for the development of new skills in the operation, familiarization, operational check-out, fault isolation and repair of AIS ATE and LRUs for the F-16 aircraft. The system provides a high-fidelity simulation of the AIS computer system, including a very detailed simulation of the software diagnostic tools used to debug complex AIS and LRU malfunctions, and a low-fidelity simulation, using digital multimedia images, of the four ATE station types of the AIS. The simulator uses actual AIS test data, obtained by using a data capture utility, to drive a simulation of the test equipment for 63 different malfunctions of both AIS equipment and aircraft LRUs. In addition, a courseware development utility provides the capability to create and modify the simulation presented at the computer bay and the multimedia simulation without having to modify trainer software.

This paper will provide an overview of the AIS-MPT software design, a description of the orchestration of the real-time simulation software and the multimedia presentation of test equipment, and an example of the unique development of "simware" materials that define student exercises.

ABOUT THE AUTHORS

Scott D. Royse is manager of the Training Systems Development Section at Southwest Research Institute. He has over 12 years of experience in systems engineering projects for automotive and missile electronics, and has developed simulator training systems for military and industrial clients. He has managed many training device development efforts including the Combined Arms Staff Trainer for the U.S. Marine Corps. He has a BSEE from Texas A&M University.

Kevin J. Miller is a senior staff member of the Training Systems Development Section at Southwest Research Institute. He has over 14 years of experience in the development of real-time software for automatic test equipment and simulator equipment for military clients. He has a BS degree in Computer Science from the University of Texas at San Antonio.

Charles J. Zinsmeyer is a Senior Research Analyst in the Instructional Systems Section at Southwest Research Institute. He has over seven years of experience in developing training systems and simulators for military and industrial clients. He has a BS/CS degree from St. Mary's University in San Antonio, Texas.

DESKTOP SIMULATION FOR AVIONICS MAINTENANCE TRAINING

Scott D. Royse
Kevin J. Miller
Charles J. Zinsmeyer
Southwest Research Institute
San Antonio, Texas

INTRODUCTION

The F-16 Avionics Intermediate Shop (AIS) Maintenance Procedures Trainer (MPT) will provide a training environment for the development of new skills in the operation, fault isolation, and repair of AIS automatic test equipment (ATE) and F-16C/D aircraft avionics line replaceable units (LRUs). The trainer, designed to teach basic (3-level) and advanced (7-level) F-16 avionics maintenance courses to 300-350 students per year, presents exercises based on data captured during troubleshooting and repair activities of actual malfunctions experienced in the field. This data provides the baseline for a full-fidelity simulation of the test station computer bay and the ATLAS test language utilities. Students use the same technical order procedures that they will apply in the field to conduct troubleshooting and repair procedures using the test bay computer, and a digital video simulation of station test equipment and aircraft LRUs. The trainer is initially supplied with 63exercises based on malfunctions that require various levels of complexity in terms of diagnostic procedure and provides complete "simware" development utilities for the authoring of new exercises. The unique combination of real-time simulation programs and multimedia-based simware of the AIS-MPT provides a data-driven, desktop simulation for a training task previously addressed only by computer-based training (CBT).

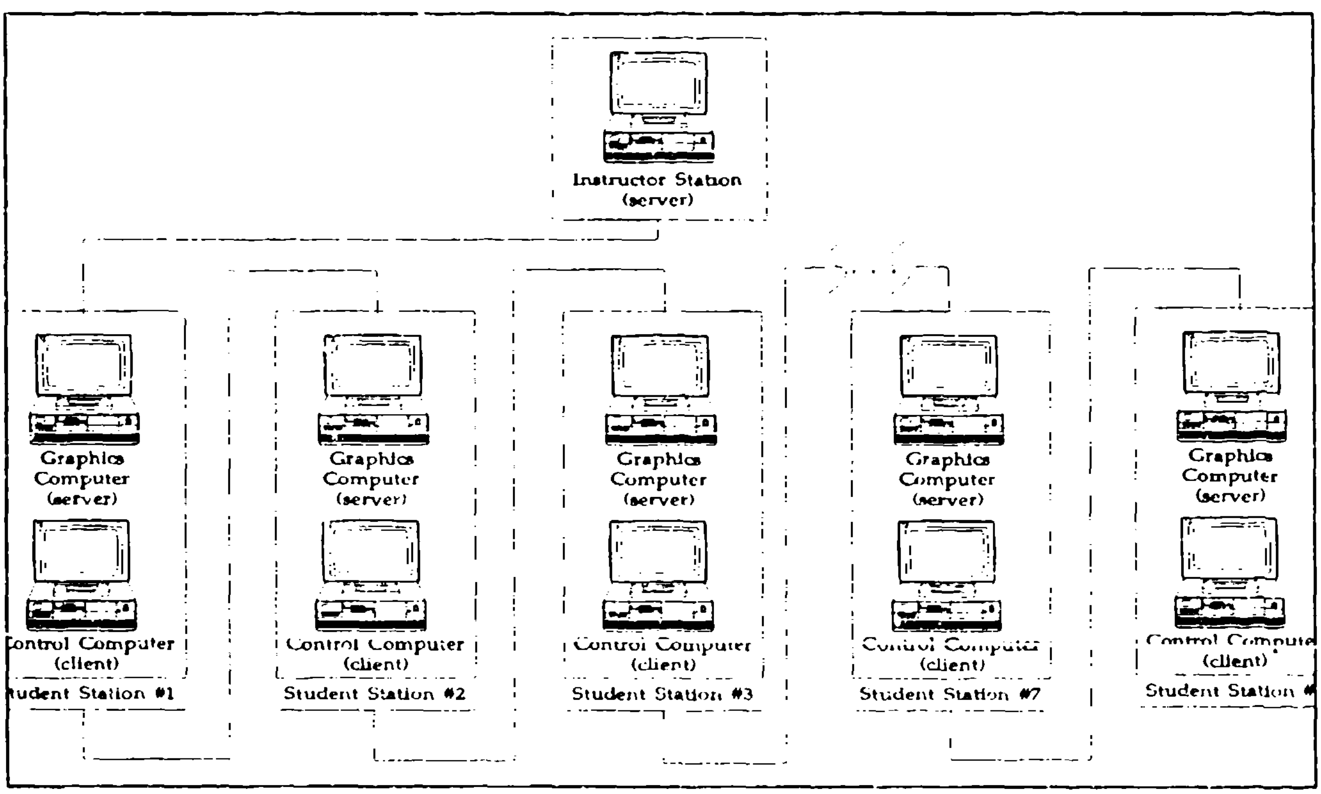


Figure 1 System Overview

SYSTEM OVERVIEW

The trainer, shown in Figure 1, consists of 19 IBM-AT compatible computer systems, including a stand-alone, dual-computer courseware development station (CDS) and a local area network containing an instructor station (IS) and eight dual-computer student stations (SS). At each SS, the trainer provides the student with a simulation of the F-16C/D AIS ATE. Four ATE station types are simulated: the computers and inertial (CI) station (Figure 2), the displays and indicators (DI) station, the pneumatic and processors (PP) station, and the radio frequency (RF) station. All stations incorporate an identical computer bay; however, each station differs in the ATLAS test software executed at the computer bay, the test equipment mounted in the equipment bays, the LRUs tested by the station, and the common test equipment used to isolate test station faults.

The physical and operational fidelity of the simulation for each ATE station and each LRU varies depending on the stated specification requirements. In simple terms, the requirements are that the physical fidelity of the simulated station computer bay shall be "high" and the physical fidelity of the test equipment bay, test accessory equipment, and LRUs shall be "low." Thus, each SS provides a full-fidelity physical mock-up of the ATE computer bay; and a multimedia simulation of the test equipment bay, the LRUs, and common test equipment is provided through the display of photographic images on a large color monitor. At the simulated computer bay, the student interacts with the equipment and ATLAS programs using the keyboard and station control panel (SCP) as he would at the actual stations. At the image monitor, the student interacts with equipment by touching the surface of the monitor while viewing equipment photographs and operator menus. The operational fidelity of the simulation is driven by a requirement to provide training exercises for the repair of 63 pre-defined ATE and LRU malfunctions. During each exercise, the student has the "freeplay" to perform maintenance actions that are not in accordance with the ideal troubleshooting path. However, the trainer does monitor student performance and will "freeze" the exercise when the student has significantly deviated (as defined by the exercise author) from the ideal path.

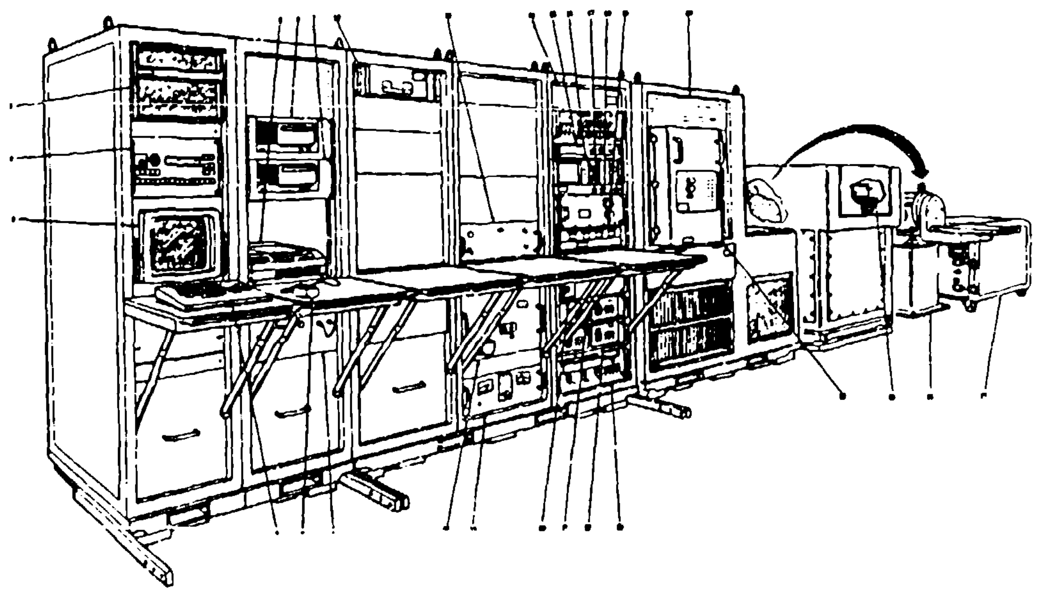


Figure 2 AIS CI Station Graphic

AIS-MPT SOFTWARE DESIGN

Overview

Trainer requirements state that the simulation of ATLAS program execution at the AIS computer bay shall be provided through the replay of observable data captured from an actual ATE computer bay. In addition, a courseware developer shall have the capability to modify the simulation presented at the computer bay and the simulation presented on the image monitor without having to modify trainer software. Thus, the software at the the SS has been designed as shown in Figure 3 to use data contained in disk files to drive the simulation. To distinguish between trainer functions provided by software and functions provided by data, data has been organized into "exercises" and "simware." Exercises monitor student performance during a training situation and provide instructional feedback to the student. Simware provides a simulation of ATLAS program execution at the computer bay and the visual simulation of LRUs and test equipment on the image monitor. Simware includes the following types of data:

a. ATLAS test language simulation authoring files that control the presentation of observables at the simulated AIS computer bay. These files incorporate captured computer bay observable data for the bay's CRT terminal, printer, and station control panel (SCP).

b. Visual simulation authoring files that control the simulation of test bay equipment, LRUs and common test equipment through the presentation of photographs/menus on the image monitor.

c. Image files that contain a digital representation of a 35mm slide or a motion sequence.

Courseware Development Station (CDS)

Ada and commercial off-the-shelf (COTS) software at the CDS as shown in Figure 4 provide a development environment, using a pull-down menu interface, that allows a courseware developer to maintain exercises and simware to keep the trainer concurrent with the F-16 AIS ATE and LRUs. To provide a development environment that

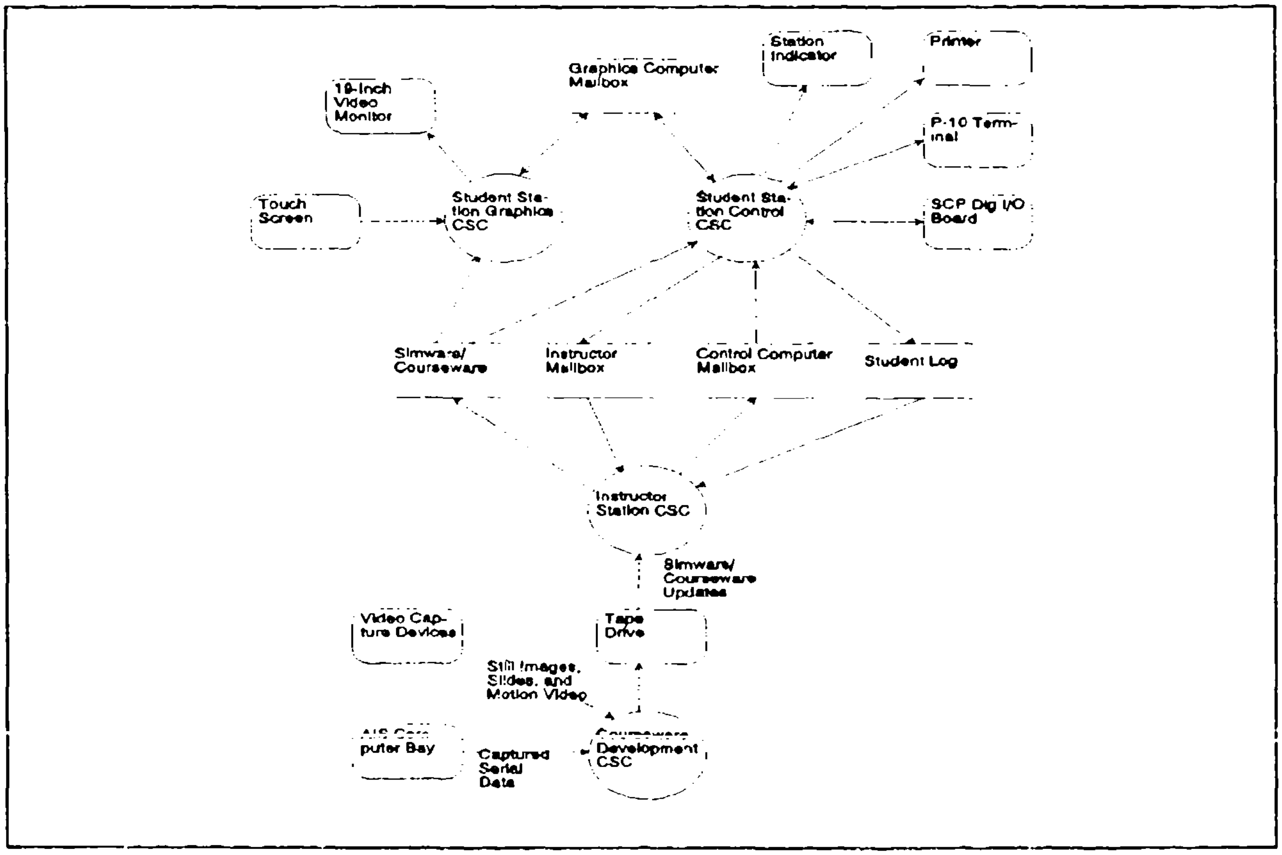


Figure 3 Software Design Overview

could be used by someone with little programming experience, three authoring systems were developed: an Exercise Authoring System (EAS), an ATLAS Simulation Authoring System (ASAS), and a Visual Simulation Authoring System (VSAS).

These tools are described as follows:

- **Exercise Authoring System (EAS).** A text editor is used to create an exercise file that initializes the simulation and defines the ideal troubleshooting procedure. Each step in the procedure "waits for" the student to perform a specific maintenance action (e.g., run an ATLAS test, reseat a circuit card) that will transition to the next step. In each step, the author can test student actions and display instructional feedback messages. A compiler is provided that checks EAS files for syntax errors and converts the file into an efficient format for execution at a student station.

- **ATLAS Simulation Authoring System (ASAS).** A text editor is used to create ASAS files that control the presentation of observables at the full-fidelity computer bay. Commands allow the author to display text on the terminal, the printer, and the SCP display in response to student actions. Input processing commands allow the author to easily simulate the characteristics of ATLAS test control menus. A compiler is provided that checks ASAS files for syntax errors and converts the file into an efficient format for execution at a student station. To automate ASAS development, the CDS provides a program to capture observable data from an AIS station. Capture requires the author to connect cables to AIS station serial ports, then exercise all ATLAS functions targeted for simulation. The capture program stores data being transmitted to the station terminal and station control panel in temporary disk files. After capture is complete, the author executes a program that converts the captured data files into ASAS authoring command files. Approximately 75 percent of the ASAS authoring process can be automated by the capture software.

- **Visual Simulation Authoring System (VSAS).** A text editor is used to create visual authoring files that control the presentation of images and menus on the image monitor. Commands allow the author to define pull-down menus,

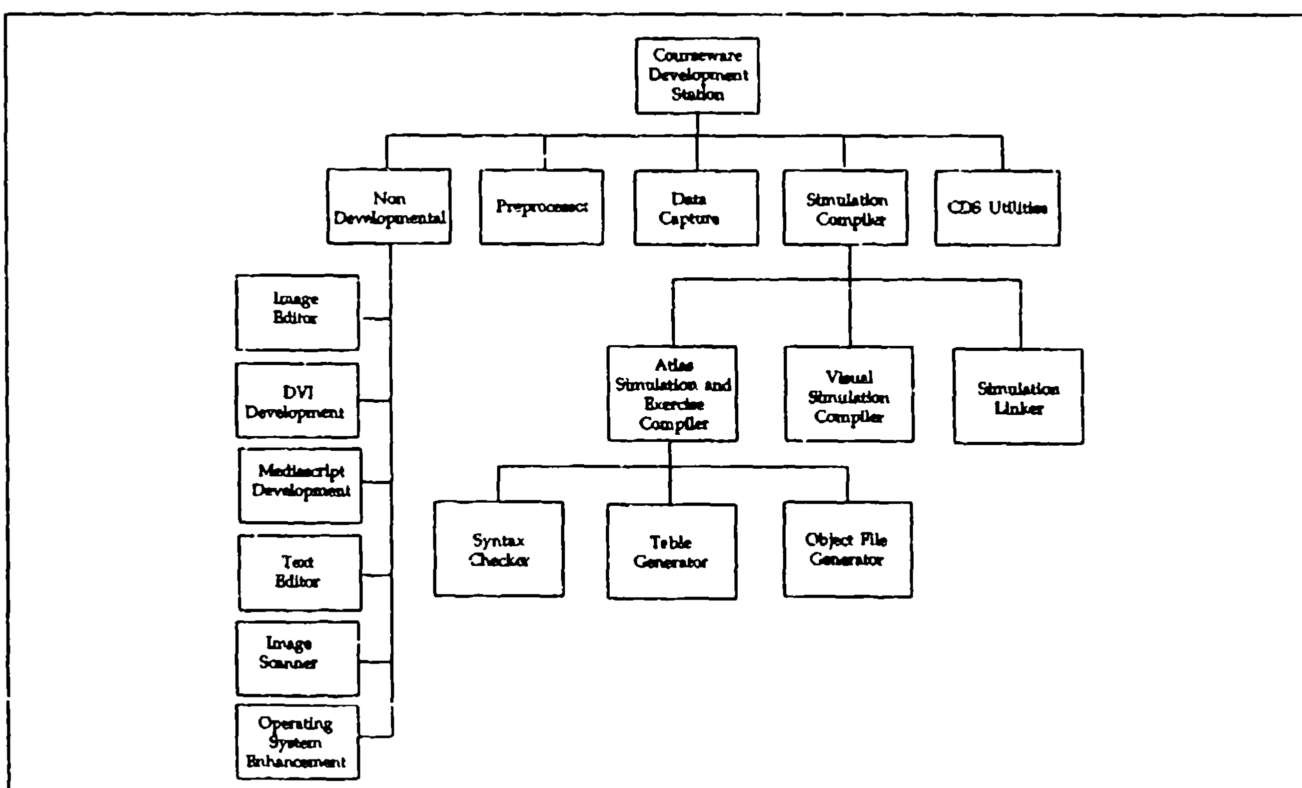


Figure 4 Courseware Development Station Software

define active touch areas, and display menus/images in response to student actions. A compiler is provided that checks the VSAS files for syntax errors and converts the file into an efficient format for execution at a student station. In support of VSAS development, the CDS provides the capability to digitize 35mm slides and motion sequences stored on SVHS tape. A COTS program is used to convert slides to image files and then edit the image files. Edit capabilities include cut/paste from multiple images, sharpening, blurring, color changes, and text overlays.

The application of these tools to the authoring of exercises, and of ATLAS and visual simware to the creation of new training scenarios, is described below.

Student Station (SS)

Software at the SS, illustrated in Figures 5 and 6, executes on two different computers and provides a simulation of the ATLAS operating system and an execution environment for simware and exercises. The primary SS computer, the control computer, executes Ada code and serves as an interpreter for EAS and ASAS files. The secondary SS computer, the graphics computer, executes 'C' code and serves as interpreter for VSAS files. Intel's Digital Video Interactive (DVI)® hardware/software is used to present images and motion sequences for a real-time simulation.

Although the ASAS and VSAS files are executed on separate computer systems, the simulation must perform as an integrated system. Student actions on the image monitor directly affect ATLAS program execution. Likewise, ATLAS test execution has a direct effect on image monitor displays. In addition, the EAS needs access to ASAS and VSAS execution to monitor student actions. Thus, the authoring systems use a common pool of variables to communicate with each other. Maintenance of the common variable pool and communications to the IS are implemented using a COTS peer-to-peer network.

Instructor Station (IS)

Ada software at the IS, shown in Figure 7, provides database maintenance functions and a student station monitor/control environment using

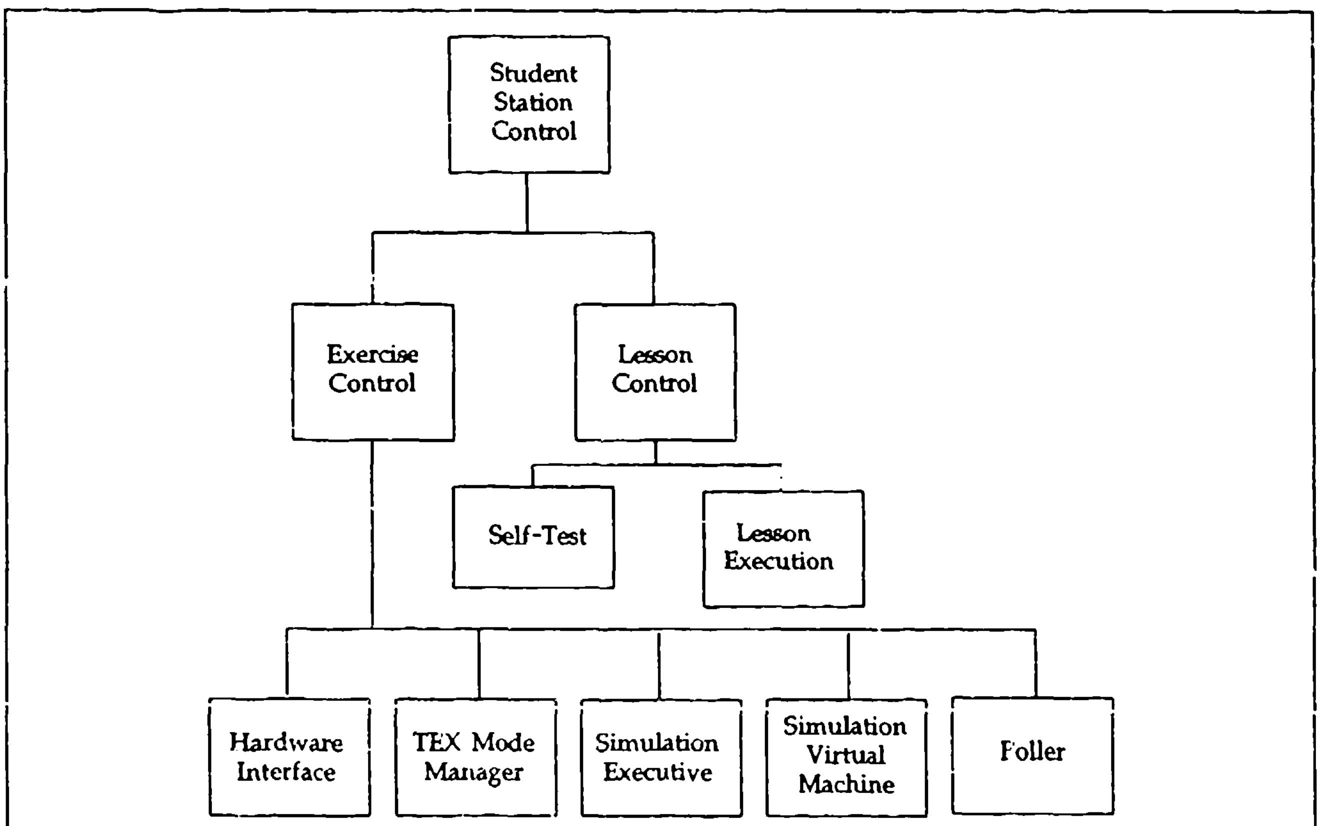


Figure 5 Student Station Control Software

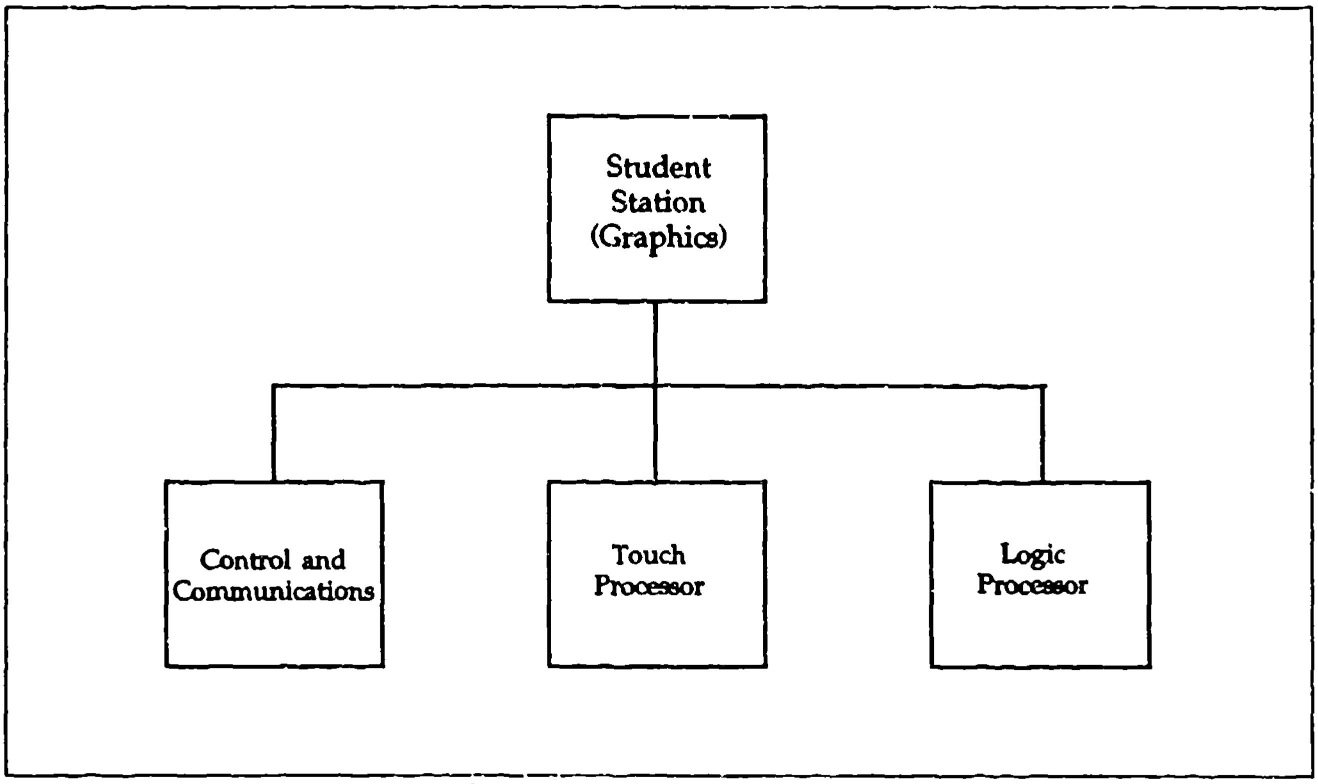


Figure 6 Student Station Graphics Software

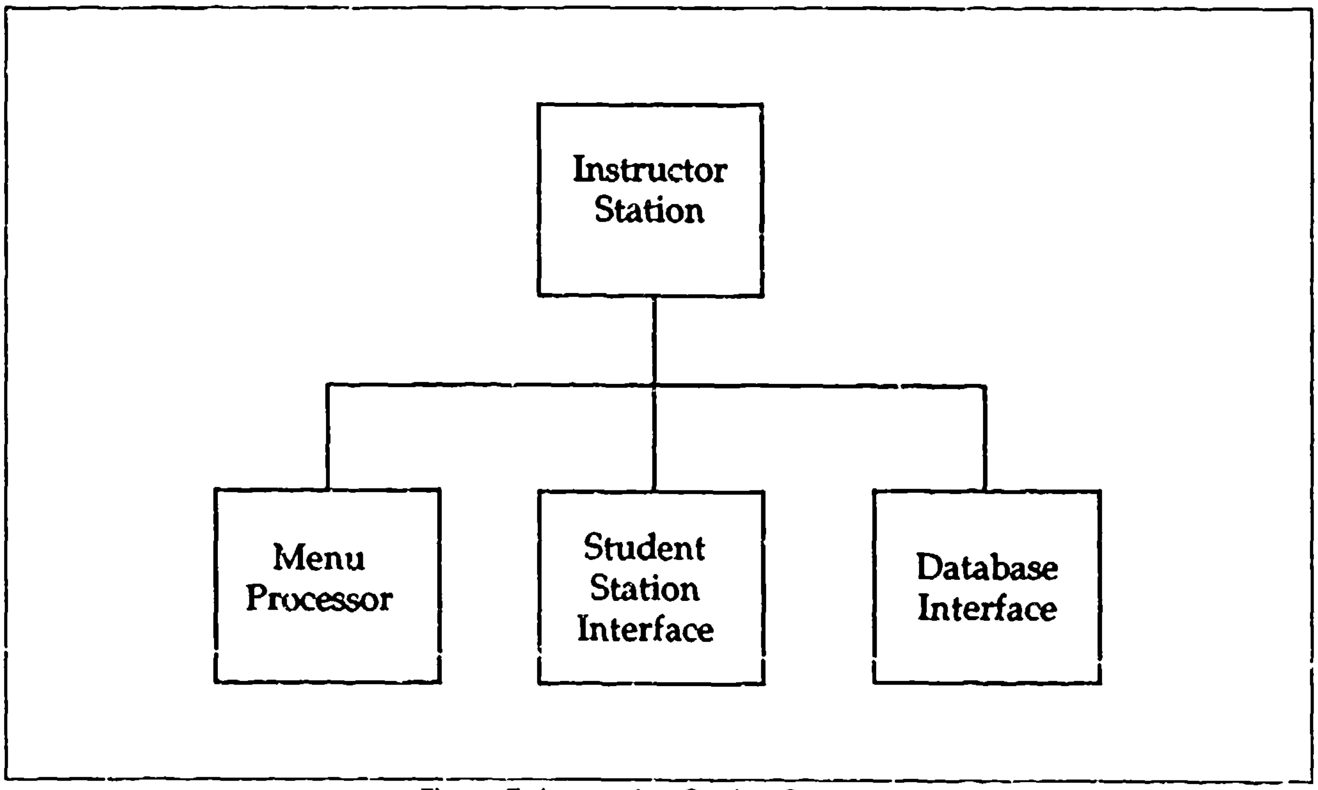


Figure 7 Instruction Station Software

a pull-down menu interface. To aid in the development of student curriculum, a lesson database is maintained at the IS. Lessons specify up to five exercises that a student will perform during a day of trainer use. In the student database, students are organized into classes and assigned an identification number. At the start of each day, an instructor assigns each student a lesson at the IS. At the SS, a student logs into the trainer with his identification number and the trainer presents the assigned lesson to the student.

Associated with each lesson conducted at the student station is an activity log. Student maintenance actions and exercise step events are recorded in the log to assist the instructor during student evaluation. From the IS, the instructor can browse through an active log for any of the eight student stations. After lesson completion, logs are archived in the student database to provide a permanent record of student performance. The IS also provides the instructor with a display indicating the status at all eight student stations. Status includes station state (e.g., idle, freeze), and a performance summary for each student (e.g., time in exercise, number of errors).

SOFTWARE-SIMWARE INTERACTION

Software at the SS is hosted on two computers that work together as an integrated system to provide a simulation of AIS equipment operation. The computers communicate using the local area network system to send and receive messages in real time. The SS control computer is connected to the simulated AIS operator's terminal, SCP, and printer. The control computer executes software written in Ada that.

(1) provides the student with a simulation of the computer bay ATLAS operating system at the terminal,

(2) presents an ATLAS program execution simulation on the terminal, SCP, and printer controlled by ASAS authoring files, and

(3) monitors student performance in accordance with EAS authoring files.

The SS graphics computer uses DVI® hardware/software to present images and motion sequences of aircraft LRUs and AIS equipment items on a 20-inch monitor with touch screen. The graphics computer executes software written in 'C' that:

(1) provides a trainer-unique user interface for students/instructors, and

(2) presents the visual simulation controlled by VSAS authoring files.

Presentation of student exercises requires the current revision of exercise and simware authoring files to reside at each SS. When a student logs in at a SS, the IS downloads a message defining the student's assigned lesson, and the first specified EAS file is executed by the control computer. File commands initialize the simulation by specifying the AIS station and equipment malfunction to be simulated. EAS execution then proceeds to the first exercise step and waits for the student to perform the required maintenance action. The AIS computer bay simulation begins by presenting the student with the ATLAS operating system prompt on the AIS operator's P-10 terminal. When the student enters an "EXECUTE" command, the appropriate ASAS authoring file is executed on the control computer to simulate ATLAS program execution. The control computer is able to execute EAS and ASAS files in parallel using Ada's tasking capabilities.

After the EAS file has initialized the simulation, the control computer signals the graphics computer to begin the visual simulation. The visual simulation begins by executing the main VSAS file for the specified station. Visual simulations always begin with an image of the entire station equipment bay displayed on the monitor. The student selects a piece of equipment to manipulate by touching it. Equipment can be extended from the bay, covers can be removed, and internal components can be adjusted, reseated, substituted and replaced.

Although the ASAS and VSAS files are executed on separate computers, the simulation performs as an integrated system. Student actions on the image monitor directly affect ATLAS program execution. Likewise, ATLAS test execution has a direct effect on image monitor displays. Together, the two coordinated presentations provide a full-fidelity simulation of the AIS computer bay, and a digital multimedia simulation of aircraft LRUs and AIS station equipment.

SIMWARE DEVELOPMENT

An important characteristic of the AIS-MPT is the inherent capability to add new training scenarios, based on data-driven simulation, without changing the simulation software. Although simware development is a complicated process, the primary skill required is that of AIS subject matter expertise, not software development. New training scenarios are created according to the process shown in Figure 8 by using the EAS, ASAS, and VSAS authoring tools described above to author exercises, ATLAS simulation files, and visual simulation files.

Exercise Authoring

The development cycle for creation of a new student exercise that will train AIS equipment fault isolation procedures begins with exercise authoring. An exercise monitors student performance during a training situation and provides instructional feedback. Exercise authoring consists of identifying the sequence of actions that constitute successful isolation and correction of a fault condition, identifying how much deviation from the 'optimal' troubleshooting procedure will be allowed, and identifying what instructional feedback will be provided in response to student actions.

For example, an AIS subject matter expert (SME) may begin the process of exercise authoring by selecting a shop replaceable unit (SRU) that will be listed in a confidence program error message as the probable cause of station failure (PCOF). The SME will then run the fault scenario on an actual AIS station to develop an Exercise Description (ED) document containing each troubleshooting step required for fault isolation and correction. The

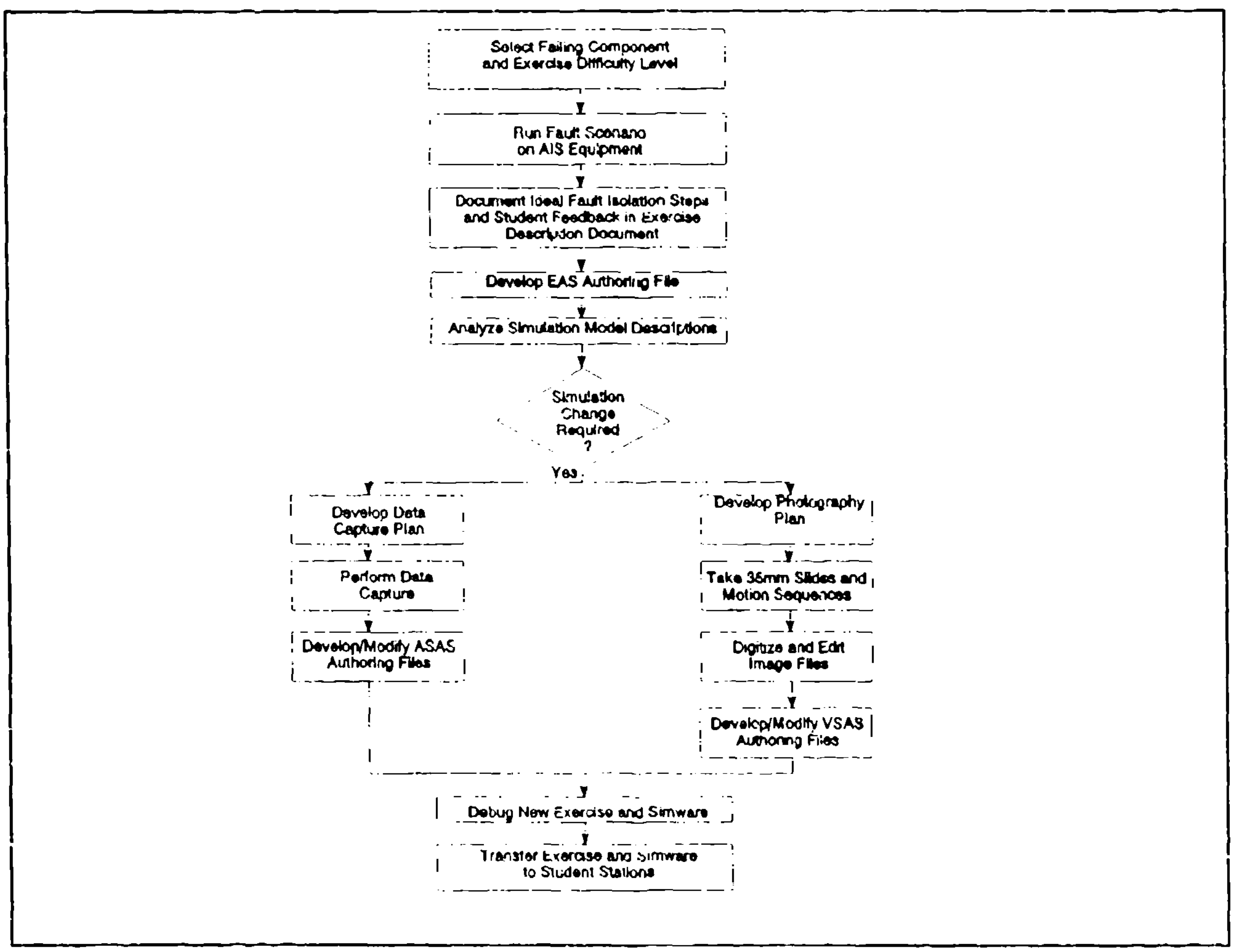


Figure 8 Simware Authoring Process

SME then adds to the Exercise Description document the instructional feedback parameters that define how the trainer will respond when the student deviates from the optimal path of corrective action. Typically, the exercise will allow the student to perform up to three significant actions that are not the next troubleshooting step of the ideal procedure before instructional feedback is provided. However, the exercise author may elect to provide feedback immediately on deviation from the preferred procedure, or may elect not to provide any instructional feedback at all. The exercise author may also choose to provide instructional feedback upon completion of each step in the troubleshooting procedure.

Once the Exercise Description document has been completed, the EAS text editor is then used to create an exercise file for execution on the Student Station Control computer.

Simulation Authoring

Once the exercise authoring is completed, the existing Simulation Model Description (SMD) documents must be analyzed to determine if simulation authoring is required to support the exercise. The SMD documents describe in detail how the simulated computer bay, test equipment and LRUs respond to specific student actions in order to provide an accurate simulation of actual equipment. For new exercises, it is likely that new ATLAS observables will be needed, thus requiring data capture activities. It is also likely that new digital image materials will be required for the visual simulation of test equipment items. Once the SMD modifications are complete, ATLAS program data capture and equipment photography plans are developed.

The flowchart shows that the ASAS and VSAS developments can be performed in parallel. ASAS development begins with data capture of actual AIS operating data performed in accordance with the data capture plan. ASAS authoring files to be modified are checked out of the configuration management (CM) system, and the data capture results are integrated into the authoring files using the ASAS development environment. The main authoring task at this point is the control logic associated with simulation of the computer bay, and the main skill required of the simware author is that of AIS subject matter expertise.

In a similar r ▸ er, VSAS development begins with photography , formed in accordance with the photography plan. The resulting 35mm slides and motion sequences are digitized and slide image files are edited to produce the final still frame images. The VSAS is then used to define active touch areas and pull-down menus, and to add the control logic associated with visual simulation of the test equipment and LRUs. Once again, the primary skill involved on the part of the author is that of subject matter expertise. Finally, existing ATLAS and visual simulation files are checked out of the CM system and modified to display the new images and allow the student to perform additional maintenance actions.

Debug and Validation

The final stage of exercise development is the debug and validation of new and modified authoring files at the CDS. This is accomplished at the CDS by selecting the debug state from pull-down menu choices and selecting the exercise to debug. The CDS is temporarily reconfigured to operate as a student station executing the new exercise. Simulation or exercise errors that are encountered during execution may be corrected by returning to the CDS development environment to effect the necessary changes. Exercise validation is accomplished by comparing proper exercise execution with the ED and SMD documents, and with actual test station operation. Once all debug and validation activities at the CDS are complete, the author transfers the modified files to the IS using tape media. The IS is then used to (1) transfer EAS and ASAS files to each SS control computer, and (2) transfer VSAS authoring files to each SS graphics computer.

CONCLUSION

The F-16 Avionics Intermediate Shop (AIS) is an expensive and extremely complicated device combining aircraft components under test with an array of electronic test equipment and sophisticated software diagnostic tools. Furthermore, as the hardware and software configuration of the aircraft changes, it impacts the test equipment and the procedures for troubleshooting and maintenance of aircraft components. The AIS Maintenance Procedures Trainer addresses this situation by providing a desktop simulation of the AIS equipment, along with the tools required for avionics maintenance experts to update and maintain existing training exercises, and to provide new training exercises without changing the simulation software of the trainer.

FROM AN INTELLIGENT JOB AID TO AN INTELLIGENT-COMPUTER-AIDED-TRAINING SYSTEM: TRAINING APPLICATIONS OF THE INTEGRATED MAINTENANCE INFORMATION SYSTEM (IMIS)

Leo Gugerty, Kimberly Hicks, William Walsh
Mei Technology Corporation
San Antonio, Texas

ABSTRACT

The goal of the project described here is to investigate the training uses of the Integrated Maintenance Information System (IMIS). IMIS is being developed by the Air Force's Armstrong Laboratory as a job aid (using automated tech-order data) for aircraft maintenance technicians who are performing duties on the flightline. This project involves conducting an analysis of current Air Force maintenance training practices, developing a prototype of how IMIS can be used in Air Force training at Technical Training Centers and operational bases, and demonstrating this prototype in a realistic training situation.

In this paper, we first describe the IMIS system and briefly report the results of our analysis of current Air Force maintenance training practices. Then, we suggest a general procedure for planning how to add training capabilities to job aids, and describe how we followed this procedure in planning the IMIS training system. Finally, the prototype IMIS training system is described.

The most effective use of IMIS in training would be as an intelligent simulation environment. In terms of the content of training, the prototype we are developing will explicitly train novice technicians (skill level 3) in some of the key strategies and knowledge used by expert aircraft troubleshooters. In terms of training methods, the prototype will follow cognitive-apprenticeship-training principles and will be designed to stimulate collaborative learning and discussion. It will include some aspects of an intelligent tutoring system, such as limited student modeling and adaptive instruction.

ABOUT THE AUTHORS

Dr. Leo Gugerty is a cognitive psychologist in the Training Technology Division of Mei Technology Corporation. His research focuses on complex problem-solving skills (such as troubleshooting) and effective interventions to train these skills. He has a Ph.D. in Cognitive/Experimental Psychology from the University of Michigan, Ann Arbor, and has held research positions at Educational Testing Service, Lockheed Engineering and Sciences Co., and the University of Houston.

Kimberly Hicks is an Industrial Psychology Specialist in the Training Technology Division of Mei Technology Corporation. Her work is concentrated in the areas of training systems analysis, and survey development and analysis. She has an M.A. in Industrial Psychology from St. Mary's University, San Antonio, TX.

William J. Walsh has been involved in the design and development of training systems, and researching training technology issues for over 15 years. He has worked on and managed programs involving various implementations of training technology, including computer-based training, multimedia applications, intelligent computer-assisted training, simulations of maintenance and troubleshooting, and distance learning among others. Recently his concentration has been on technological applications to reduce instructional development time and increase instructional quality.

FROM AN INTELLIGENT JOB AID TO AN INTELLIGENT-COMPUTER-AIDED-TRAINING SYSTEM: TRAINING APPLICATIONS OF THE INTEGRATED MAINTENANCE INFORMATION SYSTEM (IMIS)

Leo Gugerty, Kimberly Hicks, William Walsh
Mei Technology Corporation
San Antonio, Texas

INTRODUCTION

The goal of the project described here is to investigate the training uses of the Integrated Maintenance Information System (IMIS). IMIS is being developed by the Air Force's Armstrong Laboratory as a job aid for aircraft maintenance technicians who are performing duties on the flightline. This project involves conducting an analysis of current Air Force maintenance training practices, developing a prototype of how IMIS can be used in Air Force training at Technical Training Centers and operational bases, and demonstrating this prototype in a realistic training situation.

In this paper, we will first describe the IMIS system and briefly report the results of our analysis of current Air Force maintenance training practices. Then, we will suggest a general procedure for adding training capabilities to job aids, and describe how we followed this procedure in planning the IMIS training system. Finally, the prototype IMIS training system will be described.

THE IMIS JOB AID

One of the main purposes of IMIS is to integrate the wide range of information needed to perform maintenance activities and give technicians easy access to this information through a single system. IMIS is intended to integrate the following kinds of information:

- Maintenance Procedures, i.e., Technical orders (TOs)
- Diagnostic information
- Flight data
- Aircraft historical data
- Supply and management data
- Training data

Currently, only the technical orders and diagnostic information are available in the IMIS job-aid prototype. The IMIS prototype is being developed for the F-16 aircraft.

In its present configuration, the IMIS hardware consists of the Maintenance Workstation, the Portable Maintenance Aid (PMA), and the Aircraft Interface Panel. The workstation is used during the early stages of maintenance to help debrief the pilot, analyze flight and historical data, schedule the maintenance activity, and assemble a set of TOs and other information that can assist the technician during troubleshooting and repair. A portion of this information is downloaded into a memory module which is inserted into the PMA. The PMA is a rugged computer about the size of a commercial laptop. The technician takes the PMA to the flightline and hooks it up to the plane with the Aircraft Interface Panel. During troubleshooting and repair, the technician has access, via the PMA, to any TOs that are needed and to additional advice from IMIS's Diagnostic Module. The PMA can also access information from the aircraft, such as built-in-test data. The technician can use the PMA to order parts and to issue maintenance status reports via a radio link to the workstation.

One of the main components of IMIS that can be used in training is the Diagnostic Module (Cooke, Maiorana, Myers & Jernigan, 1991). This software module provides expert troubleshooting advice to the technician, based on a database of

common aircraft symptoms, the faults that could cause specific symptoms, component failure rates, and the expected results of tests and replacements. These data are used by the Diagnostic Module to provide advice during troubleshooting beyond that contained in the TOs. The database of symptoms, faults, failure rates, and test results used by the Diagnostic Module is called the Content Data Model (CDM). The CDM also contains electronic copies of the TOs needed for F-16 maintenance.

CURRENT MAINTENANCE TRAINING

In analyzing current Air Force maintenance training practices, we conducted structured interviews at Technical Training Center, Field Training Detachment, Logistics Support Training, and on-the-job training sites at Kelly AFB, TX, Hill AFB, UT, and Lowry AFB, CO. The interview data suggest that maintenance technicians especially need training in the problem-solving skills necessary to handle troubleshooting problems that are not solvable by standardized procedures (e.g., TOs). Since we felt that an effective training use of IMIS is as an intelligent maintenance simulator, we noted current uses of maintenance simulators at the bases. Maintenance simulators (part-task trainers) are currently used in the Technical Training Center and Field Training Detachment classrooms. However, there are a number of problems with their use. First, individual student access to the simulators is limited. Second, the existing simulators are hard to update and may not be used at all when they become outdated. Third, the simulators only give students problems requiring routine use of the TOs, despite the fact that technicians often face problems on the job that require knowledge beyond the TOs.

ADDING TRAINING CAPABILITIES TO JOB AIDS

Adding a training capability to a job aid presents a number of opportunities and problems that do not exist when a training system is developed from scratch. The main advantage of starting with an existing job aid lies in the often extensive job knowledge encoded in the job aid. Some of this knowledge can potentially be used in the training system. If it can, this will save some of the effort of eliciting job knowledge from human experts and encoding this in a representation that can be used by a computer. However, whether the job knowledge in a job aid can be used in a training system depends on how this knowledge is encoded or represented. If the job knowledge is represented in a form similar to the way humans think about this kind of knowledge, then the knowledge will be relatively easy to use for training. Systems with human-like representation of expert knowledge have been called *glass-box* expert systems (Burton & Brown, 1982, Anderson, 1988). The problem is that most job aids contain *black-box*, rather than glass-box, expert knowledge. That is, they represent and process job knowledge in a form very different from human experts. If a job aid uses a black-box knowledge representation, the job knowledge is much less useful in training.

Based on these considerations, we suggest the following process for planning how to add training capabilities to job aids:

1. Conduct a cognitive task analysis of the task to be trained. This will identify the subtasks necessary to perform a task at an expert level, and the knowledge and cognitive processes needed for each subtask (Hall, Gott & Pokorny, 1989).
2. Compare the knowledge representations and algorithms used in the job aid with those used by human experts. This will reveal which types of expert knowledge and cognitive processes can be trained using the existing job aid knowledge representations, and which will require more extensive development.
3. Plan which types of task knowledge and cognitive processes will be conveyed by the training system, and what training methods will be used to convey them.

Since job aids often reduce the need for training, it may seem incongruous to add training capabilities to job aids. However, for difficult tasks like maintenance, job aids will never eliminate the need for training. For example, IMIS will not be able to solve all troubleshooting problems. Given the continued need for training, integrating training and job aiding capabilities has several advantages: 1) using the intelligence of the job aid in training, 2) allowing the job aid to help train users so that they can handle the problems that the job aid cannot, and 3) providing realistic training with the job aid that is more likely

Table 1. Knowledge Used by Human Troubleshooters and IMIS

Knowledge Used by Expert Troubleshooters			Extent of Use of Human Troubleshooting Knowledge by IMIS
Knowledge Category	**Knowledge Type**	**Description**	
Declarative	Mental Models	"How-it-works" knowledge pertaining to system structure (topography), function, and behavior.	Limited
	Symptom-fault Associations	Associations between symptoms and faults. Fault probabilities.	Extensive
Procedural (Cognitive Processes)	Procedures	Step-by-step information about how to perform specific actions	Extensive
	Strategies	More general processes that coordinate the use of procedures, using mental models or symptom-fault associations.	Uses key strategies
	Coordination Processes	Very general, metacognitive processes that involve activities such as strategy selection.	Moderate

to be accepted in on-the-job training environments than stand-alone computer-based training.

CREATING AN IMIS-BASED TRAINER

Cognitive Task Analysis (Step 1)

In the following, we describe how we followed this process in the case of the IMIS job aid. Our task analysis focused on the knowledge needed for troubleshooting, as our analysis of current maintenance training suggested that troubleshooting is a high-priority Air Force training need. Our analysis of expert human troubleshooting was based on a large number of studies of troubleshooters (e.g., Rasmussen, 1981, Gott, 1988, Lesgold & Lajoie, 1991). In Table 1, we describe the types of knowledge used by expert troubleshooters (Step 1), and evaluate the extent to which IMIS uses each of these types of knowledge (Step 2).

Table 1 lists the key knowledge required for expert troubleshooting under two categories, declarative (or factual) knowledge and procedural (or skill) knowledge. The procedural knowledge (or cognitive processes) is hierarchical in nature, with the coordination processes being the most general. These processes organize the use of strategies, which in turn control the use of the more specific procedures. There is no hierarchical relationship between the two types of declarative knowledge, mental-model and symptom-fault-association knowledge. Rather, each of these types of knowledge is used by different troubleshooting strategies, as is described below.

Mental model knowledge, in the context of troubleshooting, is knowledge of how the device or system works. It includes knowledge of the internal structure (or topography) of the system, the functions of system components, and the states (or behavior) of the system. A mental model can be used to answer what-if questions about the system, such as how the failure of a particular component will affect the behavior of the rest of the system. Symptom-fault associations are remembered associations between specific symptoms (improper system

behaviors) and the internal system faults that usually cause them. When your mechanic diagnoses your car problem instantly after listening to the engine and looking at the color of the exhaust smoke, he or she is using symptom-fault associations.

Procedures knowledge consists of step-by-step information about how to perform specific actions, such as testing resistance with a multimeter. Strategies are more general processes that help organize technicians' search for the faulty component. There are two main kinds of troubleshooting strategies (Rasmussen, 1981). The first involves using symptom-fault-association knowledge to recall the fault that was found to cause a symptom in the past, and then testing for this fault. The second involves using mental-model knowledge. An example of the second type of strategy is elimination, in which the technician uses information about correct system behaviors and knowledge of system function and topography to eliminate from consideration components that can be inferred to be working. So, if your car will not start but your lights work, you can eliminate the battery as a possible cause of the problem.

Coordination processes involve metacognitive thinking, in which technicians monitor how well their strategies are meeting the various constraints of the problem (e.g., finding the fault, having a plane ready on time) and change strategies accordingly. A technician is using coordination processes when he or she decides, because of time constraints, to stop trying to isolate (locate) a specific fault and instead make a costly component replacement that will fix the problem quickly.

Comparing IMIS to the Task Analysis (Step 2)

The next step is to see how well IMIS's knowledge and algorithms match up with those used by human experts. IMIS's Diagnostic Module uses extensive symptom-fault-association and procedures knowledge. The data in the CDM on symptoms, faults, failure rates, and test and replacement results are equivalent to symptom-fault associations. The CDM data on TOs are examples of procedures knowledge.

In addition, the Diagnostic Module uses two troubleshooting strategies that are commonly used by expert technicians. The first of these is elimination. Recall that in this strategy, all components that are spanned by (i.e., lead into) a troubleshooting test yielding a successful result (a pass) are eliminated from the set of potentially faulty components. During troubleshooting, the PMA screen shows a list of tests recommended by the Diagnostic Module and a block diagram of the faulty aircraft subsystem. Different kinds of shading are used to indicate visually which components are spanned by the most highly recommended test, and which components have been eliminated based on a test result.

The Diagnostic Module also uses a version of half split (or binary search), a strategy that is used by human experts. A typical use of half split by a person employs topographic mental-model knowledge. For example, if the person knows that the current set of potentially faulty components is connected in a chain, then the half-split strategy would involve testing a component in the middle of the chain. This would ensure that, whether the test passes or fails, half of the components can be eliminated. However, the tests recommended by the Diagnostic Module do not always conform to the strict half-split strategy, because the Diagnostic Module considers other information in addition to the half-split strategy when choosing tests, information such as component failure rates and the time to perform tests and replacements.

In terms of coordination processes, the Diagnostic Module considers some of the same kinds of information that a person would in planning how to attack a troubleshooting problem and deciding what strategies to use. This includes information such as the time to perform tests and replacements, and parts availability. However, the Diagnostic Module's overall coordination processes are much different than humans. The Diagnostic Module algorithm uses an exhaustive-search approach, evaluating every fault (in its database) that could possibly cause the current symptoms, and every test that could possibly give information about the most likely fault. These faults and tests are evaluated via extensive numerical computations that are not used by human troubleshooters.

Another key difference between the Diagnostic Module and human troubleshooters is that the Diagnostic Module uses very little mental-

model knowledge, that is, very little knowledge of the system functions and topography, in its reasoning. One kind of mental-model knowledge that is used by the Diagnostic Module is knowledge of the components spanned by different troubleshooting tests. This knowledge is used by the elimination and half-split strategies. Other than this, the Diagnostic Module relies primarily on symptom-fault association knowledge, rather than mental-model knowledge.

These similarities and differences between the human and the Diagnostic Module's approach to troubleshooting will affect how IMIS can be used as an *intelligent* training system. For example, a key feature of an intelligent training system is its ability to explain its reasoning processes to students. A system's explanation capabilities will be more effective to the extent that its reasoning processes approximate those of a human expert (Anderson, 1988). The above analysis shows that, although the Diagnostic Module was not intended to simulate human troubleshooting processes, its knowledge and reasoning processes overlap considerably with those used by expert troubleshooters. The Diagnostic Module is somewhere between a *black-box* and a *glass-box* expert. In its current capacity as a job aid, the Diagnostic Module can explain its recommended tests and replacements using the following information from the CDM: the expected results of tests and replacements (symptom-fault association knowledge) and the time for tests and replacements. It would be easy for the Diagnostic Module to construct explanations based on other symptom-fault-association knowledge and troubleshooting strategies, and more difficult (though not impossible) to construct human-like explanations in term of mental-model knowledge and coordination processes.

Planning the Content and Methods of Training (Step 3)

The cognitive task analysis (Step 1) specifies the content that must be trained. In addition, comparing the job aid's capabilities to the cognitive task analysis (Step 2) gives an indication of how easy it would be to train different aspects of this content using the knowledge already in the job aid. The next step is to plan how the job aid will be used in training the content knowledge identified in the task analysis. This planning involves making choices about the instructional and assessment methods that will be used.

In the IMIS training system, we decided to use the methods of *cognitive apprenticeship training*: practice on realistic problems, coaching, fading, and collaborative learning (Collins, Brown & Newman, 1989). These methods, which will be described below, fit well with the apprenticeship methods used in on-the-job maintenance training, and have been found to be effective in teaching complex skills such as troubleshooting (Lajoie & Lesgold, 1989). Modeling IMIS's training on current on-the-job training practices will increase the likelihood of its being accepted in the flightline environment, both as a job aid and as a trainer.

IMIS can be used at each phase of the instructional process, including instructional design, instructional delivery, performance assessment, course management, and recordkeeping. In this paper, we will focus on the instruction and assessment phases.

Instructional Uses

We feel that the most effective instructional use of IMIS is as a simulator, and so we have concentrated on this use in designing the training prototype. There are two reasons for this focus. First, as a job aid, IMIS provides an interface (the PMA) that coordinates the entire maintenance process for the technician. Therefore, it would be very easy to implement realistic simulations using the PMA. Second, we feel that an IMIS simulator could be of significant help in training the most difficult skills technicians have to learn -- the problem-solving skills required for expert troubleshooting.

The apprenticeship-training literature suggests that coached practice is essential for learning complex problem-solving skills. OJT supervisors we interviewed also stressed the importance of practice in training maintenance skills. However, our data shows that in Technical Training Centers and Field Training Detachments, students have very little chance to practice their skills on aircraft or simulators. As we will describe below, an IMIS simulator can provide extensive coached practice on troubleshooting problems, thus meeting a crucial training need.

Technicians also need training in many procedures that do not involve complex problem-solving. Furthermore, some of these procedures (e.g., removing and replacing components, using a multimeter, etc.) involve perceptual judgments that are hard to teach with a simulator. This is probably the reason why OJT supervisors favor hands-on aircraft training. We feel that IMIS would be less effective at training perceptual-based procedures, at least when it is used in a stand-alone mode. Some of the problems of teaching the perceptual aspects of maintenance can be solved by using an IMIS simulator with the aircraft or other, more realistic simulators. Therefore, we hypothesize that an IMIS simulator can help teach maintenance skills in three different configurations: with the aircraft, with the existing Air Force maintenance simulators ("flatboard" simulators), and in a stand-alone configuration.

We envision two levels of detail in an IMIS simulation. The first, which we call a detailed simulation, involves augmenting the CDM with instructional information for each aircraft sub-system. This would use the full capabilities of IMIS to present maintenance training. However, authoring and updating the instructional information needed for a detailed simulation could prove expensive, especially if IMIS is used, as planned, for multiple aircraft. This led us to consider a second level of detail in simulation, wha we call generic simulation. This would use minimal authoring of instructional materials, instead relying on knowledge and data already in the Diagnostic Module and the CDM. As we mentioned previously (in discussing Step 2), in its job-aid capacity, IMIS contains considerable knowledge about aircraft systems and maintenance processes. This information overlaps considerably with the knowledge used by expert human troubleshooters. As will be described below, we feel that the information currently in the CDM is enough to create a powerful simulation environment, if it is presented to students using appropriate instructional strategies. The generic simulation would require minimal authoring and updating beyond that needed to create the CDM. Once, initially developed, the generic simulation capability would be available at little extra cost for any aircraft with a CDM that can support IMIS job aiding.

In the following, we will outline the features that could be included in a generic IMIS simulation. As mentioned above, the simulator will follow the principles of cognitive apprenticeship training.

Realistic Practice - The first feature of apprenticeship training, already discussed here, involves allowing students extensive problem-solving practice. However, if this practice is to be beneficial, the kind of problems students practice on is important. Initially, students need to practice problems that involve routine use of the TOs, as is done with the flatboard simulators. This will teach students procedures knowledge. Later, students need to practice difficult problems that cannot be solved solely by the TOs. This will require them to learn and use the other kinds of knowledge needed for expert troubleshooting. The capabilities of the IMIS Diagnostic Module potentially can allow it to solve some problems that the TOs alone cannot solve. However, IMIS's ability to provide more accurate troubleshooting advice than the TOs has not yet been convincingly demonstrated. If this ability is demonstrated, it would be possible for an IMIS simulator to generate difficult ("beyond the TOs") problems and use the intelligence in the Diagnostic Module to coach students on these problems.

Coaching - A second principle of cognitive apprenticeship involves coaching students as they solve problems. Coaching includes modeling expert problem-solving behavior and thinking, as well as giving feedback in the form of questions, hints, and reminders. We think that IMIS can provide this kind of coaching, using the knowledge in the Diagnostic Module and the CDM. Furthermore, this coaching can help students learn most of the knowledge needed for expert troubleshooting. For example, when the Diagnostic Module uses the half-split strategy to choose a test, the coach could describe this strategy to students. This is an example of modeling. Later, the simulation could give students the chance to choose their own tests and replacements, before seeing the Diagnostic Module's recommendations. At this point, the coach could determine whether the tests a student chose reflect use of half split. If they do not, the coach could remind the students of the strategy.

This example reflects a general instructional strategy of first having the Diagnostic Module model appropriate troubleshooting knowledge as the student is solving a problem. Later, the student is given more responsibility for decisions during problem solving (e.g., choosing their own tests). At this point, through questions and feedback, the coach focuses the students on the appropriate troubleshooting knowledge for the current decision. This instructional strategy could be used to instruct most of the knowledge in Table 1, in particular, symptom-fault associations, procedures, strategies, and some coordination processes. As mentioned earlier, instructing students in mental-model knowledge in a generic simulation (i.e., without adding information to the CDM) would be difficult, since the CDM contains little mental-model knowledge.

Fading - A third principle of cognitive apprenticeship is fading, whereby the coach withdraws support as the student becomes more proficient. A simple way to implement fading would be to have different levels of coaching in the generic simulation. The novice level would focus on the coach modeling and explaining the troubleshooting process, with the students having little input into the decision making. An intermediate level would allow students to have more input into decisions during problem solving, but the coach would ask questions and give feedback to ensure they were using the correct knowledge in making those decisions. The expert level would provide no coaching, allowing the students to make troubleshooting decisions as if they were using IMIS on the flightline. The instructor or the students could choose which level of coaching to use for a particular problem.

An even more effective way to implement fading would involve some student modeling. As will be described in the assessment section, a diagnostic capability could be added to the generic simulation, giving it the ability, for example, to evaluate and remember how well a student knows the half-split strategy. Given this information in a student model, the coach could determine what kind of coaching to give concerning this strategy. A student modeling capability would allow more tailoring of instruction to individual needs than simply having novice, intermediate, and expert simulations, but would be more costly to develop.

Collaborative Learning - A fourth principle of cognitive apprenticeship is to encourage collaborative problem-solving among students. This principle is already being followed in the use of the flatboard simulators in Technical School and the Field Training Detachment. Students solve problems in pairs with the instructor occasionally offering assistance or coaching. We recommend that this practice continue with IMIS simulators. In addition, other forms of collaboration are possible (Katz & Lesgold, in press). For example, students could pose problems for each other. Or, a student's performance on a troubleshooting problem could be recorded using IMIS's recording capability and later replayed for discussion by other students or the instructor. With appropriate guidance by an instructor, a classroom or learning laboratory with a number of IMIS simulators could become a rich environment for student interaction and learning.

All of the instructional features described above could be implemented in a generic IMIS simulator that uses just the CDM knowledge needed for job aiding. The detailed simulator could build on these features in a number of ways. For example, the IMIS authoring system could be used to add mental-model information to the CDM so that students could be coached in this kind of knowledge. Also, videodisk or virtual-reality capabilities could be added to the simulator. This would be of particular help in teaching students some of the perceptual aspects of maintenance activities. For example, when removing and replacing a component as part of a troubleshooting problem, students could use the videodisk screen to see what the component looks like and where it is located on the aircraft. A videodisk capability like this is already implemented for the flatboard simulators used in Technical School and the Field Training Detachment.

Assessment Uses

An IMIS simulator could perform two types of student assessment: 1) detailed diagnosis of students' strengths and weaknesses for use by an instructor, and 2) more general evaluation of student readiness for course advancement or particular work tasks. We will describe the diagnostic assessment first.

As we mentioned earlier, a generic IMIS simulator could do some simple student modeling. For example, the simulator could obtain information about a student's use of the half-split strategy by having the student choose troubleshooting tests and replacements before seeing the Diagnostic Module's recommendations (e.g., as in the intermediate level simulator described above). Using information available in the CDM, the simulator could immediately evaluate each proposed test and replacement in terms of how well it conformed to the half-split strategy. This information could be used to update a variable in a student model representing knowledge of this strategy. A similar approach could be used to model student knowledge of other strategies (e.g., elimination), symptom-fault associations, and some coordination processes. This modeling could be done as part of the generic IMIS simulator, since it relies on knowledge currently in the CDM. Only limited mental-model knowledge could be modeled using information currently in the CDM.

The information in a student model could be used in two ways. First, the simulator could use this information to tailor its coaching to individual student needs, as students work on problems. This would involve implementing the capabilities of a full-fledged intelligent tutoring system (ITS). We think this is possible, given the current configuration of the Diagnostic Module and the CDM, although the student model would be limited, in that it would not represent all the knowledge required for expert performance. A second, and less costly, way to use the detailed diagnostic information in a student model would be to print out a report describing a student's knowledge. The instructor could use this report to plan instruction, and/or students could use it to plan their studying.

A second kind of assessment that could be provided by an IMIS simulator is a general evaluation of student readiness for course advancement or job performance. For example, a supervisor could assign a student to practice a certain troubleshooting task on the simulator. When the student is ready, he or she could take a test run through the simulator on that task. The simulator could then print out a report comparing the student and the simulated expert (i.e., the Diagnostic Module) on information such as the number of tests and component replacements performed and simulated maintenance time. This information could help the supervisor decide whether the student was ready to perform this task on the flightline.

The above assessment functions could be performed by a generic IMIS simulator, with minimal changes to the CDM. A more detailed simulation could focus on functions such as assessing students' mental-model knowledge.

To summarize our ideas concerning an IMIS simulator, a generic IMIS simulator could be developed with the instructional and assessment capabilities described above (i.e., extended problem-solving practice, coaching, fading, collaboration, assessment reports, and limited student modeling). This simulator could use the intelligence and information currently available in the Diagnostic Module and CDM, and thus would not require extensive and costly authoring of instructional materials. The simulator would follow effective principles of instruction (cognitive apprenticeship) and implement some features of an ITS. We hypothesize that such a simulator would have a number of beneficial effects, including:

- increasing student practice of maintenance tasks;
- increasing student knowledge of the problem-solving skills necessary for expert troubleshooting (cf., Table 1);
- reducing training time without requiring additional instructors; and/or increasing students' proficiency levels;
- increased acceptance of computer-based training in on-the-job training on the flightline.

However, some of the troubleshooting processes used by the current version of IMIS, particularly those involving mental-model knowledge and coordination processes, are significantly different from those used by humans. Therefore, in its current state, IMIS probably could not be used to train advanced troubleshooting skills (e.g., for skill-level 7 technicians). IMIS currently offers more potential for training novice technicians, although even for training novices, the addition of some mental model knowledge to IMIS is recommended.

Using the generic IMIS simulator as a base, a more detailed simulator could be developed with

increased use of mental-model knowledge and video-based output (video or virtual reality).

CONCLUSION

In this paper, we have described our procedures for adding training capabilities to the IMIS system. After performing a baseline analysis of current Air Force maintenance training practices, we followed a three-step process of conducting a cognitive task analysis of maintenance tasks (especially troubleshooting), comparing IMIS's knowledge and algorithms to those of human experts, and developing a plan for the IMIS training system. Information from the early steps in this process was quite useful in developing the training system plan. In particular, the concept of a generic IMIS simulation followed quite closely from the cognitive task analysis and the comparison of IMIS's capability to the task analysis. The generic simulation has the potential for providing effective but low cost training by using the knowledge already available in the IMIS job aid.

REFERENCES

Anderson, J. R. (1988). The expert module. In M. C. Polson & J. J. Richardson (Eds.), *Foundations of intelligent tutoring systems.* Hillsdale, NJ: Erlbaum.

Burton, R. R., & Brown, J. S. (1982). An investigation of computer coaching for informal learning activities. In D. Sleeman & J. S. Brown (Eds.), *Intelligent tutoring systems* (pp. 79-98). New York: Academic Press.

Collins, A., Brown, J. S. & Newman, S. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. Resnick (Ed.), *Knowing, learning and instruction.* Hillsdale, NJ: Erlbaum.

Cooke, G., Maiorana, N., Myers, T., & Jernigan, J. (1991). Integrated Maintenance Information System Diagnostic Module (IMIS-DM): Version 5.0. (Report No. AL-TP-1991-0016), Dayton, Ohio:Wright-Patterson AFB.

Gott, S. P. (1988). Apprenticeship instruction for real-world tasks: The coordination of procedures, mental models, and strategies. In E. Z. Rathkopf (Ed.), *Review of research in education, Volume 15,* (pp. 97-169). Washington, DC: American Educational Research Association.

Hall, E., Gott, S., and Pokorny, R. (1990). A procedural guide to cognitive task analysis: The PARI methodology. Air Force Human Resources Laboratory Technical Working Paper. Armstrong Laboratory, Brooks Air Force Base, San Antonio, TX.

Katz, S. & Lesgold, A. (in press). The role of the tutor in computer-based collaborative learning situations. In S. Lajoie & S. Derry (Eds.), *Computers as cognitive tools.* Hillsdale, NJ: Erlbaum.

Lajoie, S. P & Lesgold, A. (1989). Apprenticeship training in the workplace: Computer-coached practice environment as a new form of apprenticeship. *Machine-Mediated Learning, 3,* 7-28.

Rasmussen, J. (1981). Models of mental strategies in process plant diagnosis. In J. Rasmussen & W. Rouse (Eds.), *Human detection and diagnosis of system failures.* New York: Plenum Press.

Acknowledgements - The IMIS trainer project is funded by Contract #F33 615-91-D-0651 from the Armstrong Laboratory (HRTC), Brooks Air Force Base, San Antonio, TX, USA; Dennis Gettman and Capt. Anne Reiners, contract monitors. We would like to thank Ron Shelfer for comments on the paper.

QUANTITATIVE CORRELATION TESTING FROM DOD PROJECT 2851 STANDARD SIMULATOR DATA BASES

Budimir Zvolanek, Douglas E. Dillard,
Joe R. Stewart, and Erwin W. Baumann
McDonnell Douglas Training Systems
McDonnell Douglas Corporation
St. Louis, Missouri

ABSTRACT

Correlation of out-the-window visual, infrared, radar, and other sensor displays must exist in local and networked simulation environments in order for users to obtain meaningful and consistent information about the simulated world. The use of a large numbers of image generators with varying capacities and the networking of simulators capable of varying degrees of fidelity underscore the need for consistent and quantitative specifications of and automated testing tools for correlation.

This paper describes the evaluation of correlation potential from databases provided by the DoD Standard Simulator Data Base Project 2851. It presents quantitative correlation testing metrics and software developed for evaluation of Project 2851 Generic Transformed Data Bases. It also describes the application of metrics to compute the degree of correspondence between simulation databases — not only for terrain elevation but aiso for feature attribute data. Our results demonstrate varying degrees of correlation potential between databases and levels of detail and verify the utility of the specifications, quantitative metrics, and automated tools to predict correlation from databases.

ABOUT THE AUTHORS

Budimir Zvolanek is a Technical Specialist in Technology Development and Integration at McDonnell Douglas Training Systems (MDTS). Mr. Zvolanek has concentrated his career on the application and development of electronic imaging systems, sensor image processing, and database development. He has been a principal investigator in image simulation and database technology development and a program manager for the DoD Standard Simulator Data Base Project 2851 contracts at MDTS. Mr. Zvolanek received his M.S.E.E. degree from Washington University in St. Louis, Missouri.

Doug Dillard is a Lead Engineer at MDTS. Mr. Dillard has concentrated his career on the design of visual and sensor simulations and the design and construction of databases for computer image generators used in aircrew training systems. He has worked on research contracts for Project 2851 and in internal research on databases for MDTS. Mr. Dillard received his B.S. degree from St. Louis University in St. Louis, Missouri, and an M.S. degree from the University of Missouri at Rolla.

Joe Stewart is a Senior Engineer at McDonnell Douglas. Mr. Stewart has concentrated his career on the design, construction, and validation of sensor databases, image processing, and machine vision. He has worked on research contracts for Project 2851 and on the design and construction of graphics for simulator instructor stations at MDTS. He received his M.S.E.E. degree from Washington University in St. Louis, Missouri.

Erv Baumann is a Senior Technical Specialist at MDTS. Mr. Baumann has specialized in the application of artificial intelligence methods to sensor simulation, machine vision, and automated planning. He has conceived and developed innovative applications of neural networks to multi-spectral image analysis and sensor simulation. Mr. Baumann received his B.S.E.E. degree from the University of Minnesota.

QUANTITATIVE CORRELATION TESTING FROM DOD PROJECT 2851 STANDARD SIMULATOR DATA BASES

Budimir Zvolanek
Douglas E. Dillard
Joe R. Stewart
Erwin W. Baumann
McDonnell Douglas Training Systems
McDonnell Douglas Corporation
St. Louis, Missouri

INTRODUCTION AND BACKGROUND

Advances in image generation technology have provided increased realism and fidelity of sensor and visual simulations. At the same time, modern aircraft cockpits have created a need to simultaneously simulate outputs from many sensors in conjunction with Out-The-Window visual and moving map imagery. Furthermore, networked simulations of multiple systems require coordinated sensor simulations which use a variety of image generators operating at different levels of fidelity and detail.

McDonnell Douglas Training Systems (MDTS), a sub-division of the Aircraft and Missile Support Systems division within the McDonnell Douglas Aerospace, has been conducting on-going R&D efforts to improve correlation in simulation environments. In Phase 1, described in a previous paper[1], definitions of correlation and algorithms to test and predict correlation "potential" from simulator databases were developed. In Phase 2, described in this paper, the definitions and algorithms were implemented in software and used to actually predict correlation potential of databases.

This paper describes the evaluation of correlation potential from databases provided by Project 2851 (P2851). It presents quantitative correlation testing metrics and software developed for evaluation of P2851 Generic Transformed Data Bases (GTDBs). It not only describes the computation of the degree of correspondence of terrain but also feature data between different Simulator Levels of Detail (SLODs) and between infrared (IR), radar, and Out-the-Window (OTW) visual GTDBs. The results verify the ability of the specifications, metrics, and tools to predict correlation potential from databases, and demonstrate varying degrees of correlation potential among the GTDBs.

Traditional Definitions of Correlation

In a previous paper[1], we described the results of a survey of correlation definitions for simulation use, using sources which were both internal and external to MDTS. The survey found that many terms are used to describe correlation, and that there is a lack of accurate definitions and objective metrics. In addition, there is no agreement on the meaning, specification, and measurement of correlation.

Need for Quantitative Measures

Simultaneous sensor and visual simulations often lead to a requirement for correlation. Although "correlation" has been used in signal processing, image processing, and communication theory to detect signals in noise and match patterns, the term has been used extensively in the simulation and training industry without accurately defining it. Meaning, specification, and method of the correlation measurement often causes users to be limited only to qualitative assessments of correlation over selected portions of gaming areas. The definitions and metrics proposed in our first paper provide techniques to automatically evaluate maximum achievable correlation over an entire gaming area and identify locations where insufficient correlation may be a problem.

Estimation of Correlation "Potential"

In the traditional database generation process, the user collects data from various sources, converts

it into a common database, and then converts the common database into run-time databases for its target computer image generators (CIGs). Each run-time database is then processed by its CIG to produce an image or input to the users. Although correlation can be measured at several points in the process, comparisons of the input data provide the best measurement of correlation, since the correlation of the outputs can be no better than the correlation of the inputs. Thus, the databases for sensors and visuals provide the basis for measuring the maximum static correlation attainable by ideal CIGs and display simulators.

A maximum correlation "potential" can be computed from the database contents without CIG-particular processing effects and simulated display distortions. More importantly, when CIG requirements, such as surface polygonalization or object culling, affect database construction and generation, such effects of such "tailoring" will be measured via database-to-database correlation.

PROJECT 2851

Project 2851 (P2851) is a research and development program chartered by the Joint Technical Coordinating Group for Training Systems and Devices. Project 2851 develops systems and standards for the efficient generation, updating, storage, and distribution of DoD simulator databases. It seeks not only to improve efficiency and lower costs of providing databases for a very large number of DoD simulators but also to improve correlation among multiple simulators and Image Generator (IG) outputs.

Project 2851 will result in a DoD Simulator Data Base Facility which will obtain standard Defense Mapping Agency products, externally generated simulation data bases, and other source material; archive and manage this data; and provide tailored data base products to DoD training simulators.

The Generic Transformed Data Base

The P2851 Generic Transformed Data Base (GTDB) is a data base which supports real-time visual, infrared and radar sensor simulation systems. The single GTDB format is a superset of data base features required for its target simulations. This single format reduces software development and maintenance costs among users of the GTDB.

The P2851 Common Data Base Transformation Program (CDBTP) generates GTDBs for specific image generators, using the terrain, culture, model, and texture data maintained at the Simulator Data Base Facility. The CDBTP uses a set of transformation parameters to determine how to tailor a GTDB. Among other things, these parameters may specify the map projection, goodness of fit for polygonal terrain, and complexity reduction.

CORRELATION TESTING DEVELOPMENT

We have defined database correlation as the degree of correspondence among data types and their contents needed by image generators. Thus, our metric for terrain elevation is the statistical distribution of terrain elevation differences between two databases; and our metric for feature data is the percentage of zero differences in cultural pixels between two databases.

Quantitative Metrics

Our quantitative metrics determine the correspondence of data among SLODs, gridded and polygonal terrain, and areal, linear, and cultural feature attributes for OTW visual, IR, and radar GTDBs.

Elevation Correlation – The elevation correlation is obtained by producing gridded elevation representations from two databases and subtracting these to obtain a grid of elevation differences. The elevation correlation is then represented by statistical measures of the differences: the average value, the standard deviation, and the range of values.

Feature Correlation–Since GTDB features overlay each other according to their rendering priority, feature correlation will be a match in feature location, orientation, size, shape, and attribution. Thus, as illustrated in Figure 1, feature correlation is obtained by subtracting composite gridded feature attributes in two databases and obtaining the percentage of zero differences.

Our feature representation uses a pointer method to reduce requirements for attribute data storage. Instead of storing all feature attributes at each grid post, the feature gridding software stores a pointer to unique sets of feature attributes, called Feature Attribute Vectors (FAVs) (Figure 2). The FAV list is common to both databases; during feature gridding,

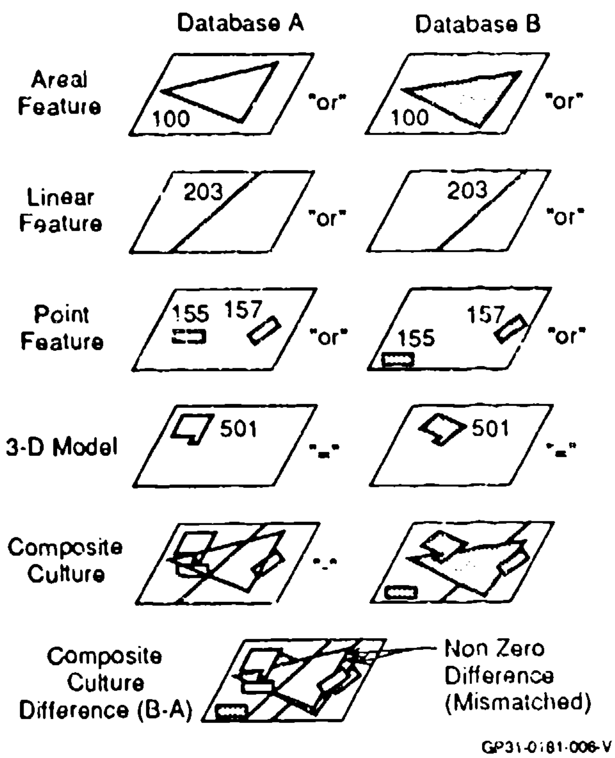


Figure 1. Cultural Feature Correlation Grid Generation

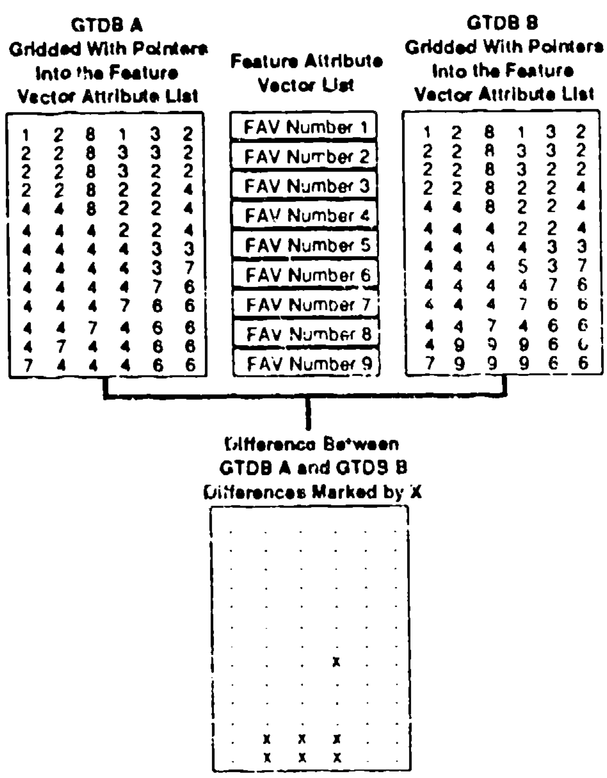


Figure 2. Cultural Correlation From Feature Attribute Vectors

current feature attributes are compared to the attributes already in the FAV list. If a matching FAV is found in the list, the associated list pointer is stored at the grid location. If none of the list FAVs matches the current feature attributes, the current attributes are added to the FAV list with the list pointer incremented and stored at the grid location.

Therefore, a tremendous reduction in feature storage is achieved by using only one FAV per feature. Cultural feature match is a percentage of zero differences in FAV pointers, or the percentage of zero differences in selected elements of the FAVs. Table 1 shows examples of attribute combinations stored in FAVs containing from 2 up to 23 attributes.

Table 1. Feature Attribute Vector Contents

Feature Attribute Vector (FAV)	FAV Contents	
FID/SMC (2)	Feature Description Surface Material	
Height (1)	Feature Height	
Infrared (7)	IR Directivity Exitance Emissivity Self-Emitter Flag	Reflectivity Absorptivity Transmissivity
Visual (6)	Visual Directivity Color Specular Flag	Hue Chroma Translucency
Radar (4)	Radar Directivity Diffuse Reflectivity Feature Onset Flag Low Level Effects Flag	
Complete (23)	Infrared Set + Visual Attribute Set + Radar Set Surface Material Feature Description Correlation Priority	 Material Subtype FID Value Feature Height

GP31-0181-10-V

Software Tools

Our GTDB processing involves three stages. Preprocessing, the first stage, is depicted in Figure 3. The second and third stages are visualization and analysis. In visualization, images of two gridded databases are compared using image processing and analysis tools. In the third stage, described below, the tools compare two gridded database representations

Figure 3 shows the preprocessing steps used for each GTDB. First, program "Read2851" reads the

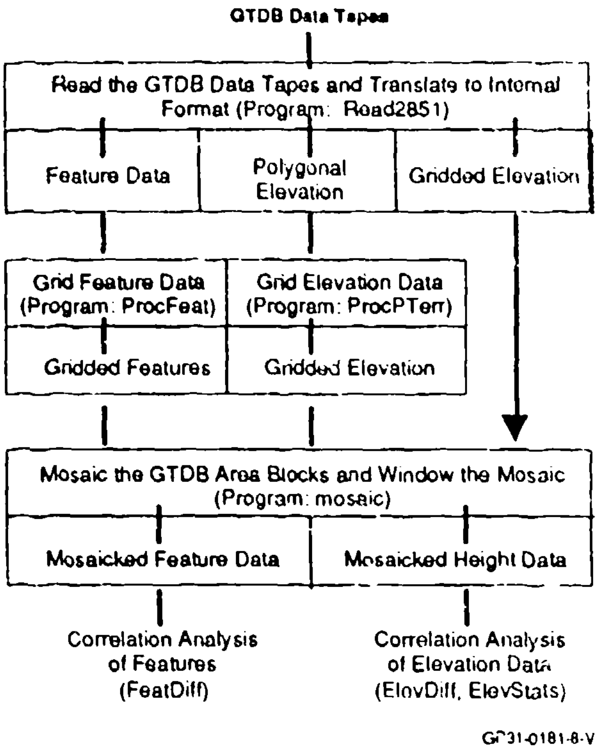


Figure 3. GTDB Preprocessing Before Correlation Testing

data tapes and transfers the data to disk files. This program produces three types of files for each area block. One file type contains contain binary representations of the cultural and feature data; another contains a binary representation of polygonal elevation data; and a third type contains gridded elevation data.

The program "ProcFeat" processes the feature data, and the program "ProcPTerr" processes the polygonal elevation data. Each program produces gridded representations of one area block. The "ProcPTerr" program produces gridded elevation data which has spacings of 3 arc seconds. The program "ProcFeat" grids areal, linear, and point features and produces feature attributes for each pixel.

The "mosaic" program combines its input files by positioning adjacent area blocks next to one another in a seamless mosaic of the database and clipping the resulting mosaic to the edges of a specified window.

The "FeatDiff", "ElevDiff", and "ElevStats" programs analyze the mosaicked gridded databases. The "FeatDiff" program reads gridded mosaics from two sources, computes the correlation, and outputs the percent correlation between its data inputs. It computes differences between FAV pointers; it does not compute differences between elements of FAVs. Figure 4 shows the two formulas which are used to compute correlation. If the gridded mosaic is totally covered with data, the formula for "Total Coverage With Valid Data" is used. If only a small part of the gridded area contains valid data, the "Small Coverage With Valid Data" is used.

For Total Coverage With Valid Data

$$\text{Correlation} = \frac{\text{Total Number of Image Pixels} - \text{Number of Mismatched Pixels}}{\text{Total Number of Image Pixels}}$$

For Small Coverage With Valid Data

$$\text{Correlation} = \frac{\left[\begin{array}{l}\text{Number of Non-Zero A-Pixels} + \\ \text{Number of Non-Zero B-Pixels} - \\ \text{Number of Mismatched Pixels}\end{array}\right]}{\begin{array}{c}\text{Number of Non-Zero A-Pixels} + \\ \text{Number of Non-Zero B-Pixels}\end{array}}$$

GP31-0181-9-V

Figure 4. Calculation of Correlation Value

The "ElevDiff" program reads gridded elevation mosaics from two databases or Simulator Levels of Detail and computes gridded differences. The "ElevStats" program computes the average value, the range, and the standard deviation of the difference.

GTDB CORRELATION TESTING RESULTS

We evaluated correlation for F-16, B-2, and Kuwait (Desert Shield) GTDBs. We measured terrain correlation between gridded and polygonal elevation representations; and we determined feature correlation as a function of database, simulator level of detail (SLOD), and multi-sensor attributes.

For a GTDB with two SLODs, SLOD1 represents more feature detail than SLOD2, effectively increasing the spatial resolution. Table 1 illustrates the FAVs used.

When discrepancies in correlation were found, the missing or badly aligned features could be found by inspecting the visual images of the gridded mosaics. For example, the F-16 LANTIRN GTDB, G000665, lacked several large areal features which are in the F-16 VISUAL GTDB, G000671. The numeric results for these GTDBs are low and reflect this disparity. Likewise, the visual comparisons of

the two LANTIRN SLODs for G000665 indicate only minor differences involving linear and point features, a fact which is reflected by the high values of correlation for these GTDBs.

Combinations of areal, point, and linear features were used in the measurement of feature correlation. When polygonal terrain data was available, measurements were also made for gridded and polygonal terrain. Since the heights of features were a small percentage of the terrain heights, no attempt was made to measure differences between composite elevations. In these GTDBs, point and linear features covered relatively small areas, and made insignificant contributions to correlation when large areal features were considered. Therefore, the formula for large features was used to compute the values presented in the following tables.

F-16 Infrared vs. Visual GTDBs

The F-16 GTDB evaluation used one visual GTDB and two LANTIRN IR GTDBs. The data bases were:

F-16-LANTIRN, G000665 (two SLODs)
F-16-LANTIRN, G000754 (one SLOD)
F-16-VISUAL, G000671 (one SLOD)

One of the LANTIRN databases, GTDB G000665, had no linear features because linear features had been expanded into areal features. The two other GTDBs, G000754 and G000671, contained areal, linear, and point features. The Northeast and Southwest corner points of the area over which comparisons were made are 39.42166666N, 118.7225000W and 39.20000000N 119.2216666W.

Features – Feature correlation between SLODs and between sensors was evaluated for all three F-16 GTDBs. The common areas of the databases were evenly gridded at intervals of 0.3 arc-seconds (about 30-foot resolution).

Table 2 summarizes the percent agreement between SLODs and GTDBs when areal, linear, and point features were gridded. The same correlation measures were obtained for the complete, visual, and IR FAVs (Table 1). When areal, linear, and point features were processed over the feature grid, the percent correlation varied between 79.9 and 99.7 percent. It is obvious from the 99.7 percent agreement between the visual GTDB, G000671, and the LANTIRN GTDB, G000754, that these two databases are nearly identical. The poor agreement between the LANTIRN GTDB, G000665, and G000671 was due to the fact that G000665 did not have several large areal features which appeared in G000671.

Table 2. F-16 GTDB Feature Correlation

Sensor-Sensor and SLOD-SLOD Correlation

Features Gridded:
Areal, Linear, and Point

Feature Attribute Vector Contains:
Complete Set or Visual Set or IR Set

	LANTIRN G000665 SLOD 2	Visual G000671 SLOD 1	LANTIRN G000754 SLOD 1
LANTIRN G000665 SLOD 1	89.98	79.97	79.96
LANTIRN G000665 SLOD 2	N/A	79.94	79.90
Visual G000671 SLOD 1	N/A	N/A	99.74

GP31-0181-11-V

Point feature correlation, as measured by the formula for small coverage, ranged from 93 to 99% for the F-16 LANTIRN GTDB 665 SLOD 1 versus SLOD 2 and GTDB 665 SLOD 2 versus the Visual GTDB 671 SLOD 1. It was 100% for the F-16 LANTIRN GTDB 754 versus the VISUAL GTDB 671.

Elevation – Terrain elevation correlation was measured between combinations of all three F-16 GTDBs. The common areas of the three databases were evenly gridded at intervals of 3 arc-seconds (about 300-ft resolution). All the gridded representations were identical. The polygonal elevation database of SLOD 1 of G000665 was nearly equivalent to the polygonal elevation database of G000754. Table 3 summarizes the results for gridded versus polygonal terrain representations.

B-2 Radar vs. Visual GTDBs

The B-2 GTDB data bases were:

B-2 VISUAL, G000774 (two SLODs)
B-2 RADAR, G000781 (one SLOD)

Table 3. F-16 GTDB Elevation Correlation

Gridded vs Polygonal Terrain

LANTIRN G000665 SLOD 1	LANTIRN G000665 SLOD 2	LANTIRN G000754 SLOD 1
A = 1 D = 12.11 R = 313	A = 7 D = 21.88 R = 328	A = 1 D = 12.15 R = 234

Legend. A = Absolute Value of Elevation in Meters
D = Standard Deviation
R = Range of Values (Max-Min)

GP31-0161-12 V

The visual database had areal, linear, and point features. The radar GTDB contained only areal and point features; the linear features had been expanded into areal features. The Northeast and Southwest corner points of the area over which comparisons were made are 32.50000N, 116.750W and 33.000N, 117.250W.

Features – Feature correlation between levels of detail and between databases was evaluated for both GTDBs. The databases were evenly gridded at intervals of 0.3 seconds, so they had 6,001 points along the northern and southern borders and 6,001 points along the eastern and western borders.

Table 4 summarizes correlation between SLODs and GTDBs when various combinations of areal,

Table 4. B-2 GTDB Composite Feature Correlation

Sensor-Sensor and SLOD-SLOD Correlation

Features Gridded:
Areal, Linear, and Point

Feature Attribute Vector Contains:
Complete Set or Visual Set or IR Set

	Visual G000774 SLOD 2	Radar G000781 SLOD 1
Visual G000774 SLOD 1	68.39	[illegible]8.27
Visual G000774 SLOD 2	N/A	98.90

GP31-0161-12 V

linear, and point features were gridded with pointers to four FAV sets. The same correlation values were obtained using four different FAVs - the complete, visual, IR, and radar attribute sets (Table 1). It shows that when areal, linear, and point features were processed over the feature grid, the percent correlation varied between 68 and 99 percent.

Point feature correlation, as measured by the formula for small coverage, was 97% for the B-2 GTDB, VISUAL SLOD 2 versus the B-2 Radar GTDB.

DESERT STORM (KUWAIT) GTDB

Feature correlation was evaluated between the two SLODs available in the GTDB. Mosaics which encompassed the entire 1-degree by 1-degree database were used for all evaluations. The database was evenly gridded at intervals of 0.6 arc-seconds, so the mosaics had 6,001 points along the northern and southern borders and 6,001 points along the eastern and western borders. The best correlation, 99.1%, was obtained when the FAV contained only feature height. For other FAVs, the correlation depended on the contents of the FAV; and correlation of the composite of Areal, Linear, and Point Features ranged between 93 to 94 percent.

CONCLUSIONS AND FUTURE DIRECTIONS

Our investigations have resulted in the following conclusions:

1. We have verified the validity of automatic correlation testing between two GTDBs or SLODs by visual inspection of the databases.
2. We have measured varying degrees of correlation between P2851 GTDBs. Very low correlation values indicate that large features are missing in a SLOD or GTDB. Values greater than 90% indicate that small features are missing or that there is some misalignment of features.
3. We have observed that several differently constructed Feature Attribute Vectors produced nearly equivalent feature correlation values.

In the future, we will implement automated tools for database texture and model correlation testing. We will also investigate tools for correlation testing from image generator video displays and relate the results to the correlation "potential" from databases.

ACKNOWLEDGMEN

The authors would like to thank Gene Clayton, the P2851 Program Manager at PRC, Gene Naccarato, and Ted Zyla for their support and commitment to the success of this effort.

REFERENCE

1. Zvolanek, B. and Dillard, D. "Database Correlation Testing for Simulation Environments", Proceedings of the 14t. /ITSEC Conference, November 1992., pp. 867-' 75.

SCALEABILITY TOOLS, TECHNIQUES, AND THE DIS ARCHITECTURE

Daniel J. Van Hook, Deborah J. Wilbert, Richard L. Schaffer
Loral Advanced Distributed Simulation
Cambridge, Massachusetts

Walter Milliken
BBN Systems and Technologies
Cambridge, Massachusetts

CDR Dennis K. McBride, PhD.
Advanced Systems Technology Office
Advanced Research Projects Agency

ABSTRACT

"Scaleability" is the concept of dramatically increasing, or scaling up, the number of simulated vehicles (and other "entities"), human participants, and geographically dispersed sites involved in a Distributed Interactive Simulation (DIS). ARPA is developing a scaleability toolset which will allow the development and evaluation of DIS exercise scenarios, scaling and network algorithms and techniques, and network topologies in a simulation environment with the goal of developing solutions to the DIS scaling problem. In addition to giving an overview of the architecture and capabilities of the scaleability toolset, this paper presents an example of, and representative results from, employing the toolset to characterize a specific promising scaling technique, the grid-based geographic filtering algorithm.

ABOUT THE AUTHORS

Daniel J. Van Hook is a senior scientist with Loral Advanced Distributed Simulation. He holds an S.M. in Electrical Engineering and the Electrical Engineers' degree, both from the Massachusetts Institute of Technology, and B.S. from Rutgers University, also in Electrical Engineering. Mr. Van Hook has worked in the area of distributed simulation for the past eight years, including the SIMNET Large Scale Simulation project. His focus has been on simulation system architectures with an emphasis on software development. He has participated in developing vehicle, sensor, and radio simulations, digitized voice communication systems, network interfaces and protocols, graphics algorithms and system software.

Richard L. Schaffer is a scientist with Loral Advanced Distributed Simulation. He holds a B.S. in Physics from the Massachusetts Institute of Technology. Richard has worked in the area of distributed simulation for the last eight years, including the SIMNET Large Scale Simulation project. Mr. Schaffer has been actively involved in the Distributed Interactive Simulation standards process since its inception in August, 1989. As part of that effort, he has chaired the DIS Articulated Parts and Radio Communication subgroups.

Deborah Wilbert is a scientist with Loral Advanced Distributed Simulation. She holds a Bachelor of Science in Computer Science and another in Science and the Humanities with a concentration in writing, both from the Massachusetts Institute of Technology. Ms. Wilbert has worked in the area of distributed simulation for the past five years, including the SIMNET Large Scale Distributed Simulation project. She has a background in network management, has developed a simulator manager, and is currently a member of the Simulation Management subgroup working on Distributed Interactive Simulation standardization. She has also participated in the development of C^3 simulation systems, semi-automated forces, and protocol definitions.

Walter Milliken is a senior scientist with BBN Systems and Technologies' Network Technology Department. He holds an M.S. in Electrical Engineering from Stanford University and B.S. degrees in Electrical Engineering and Computer Science, both from Washington University in St. Louis. Mr. Milliken has worked in the areas of computer networking and parallel processing systems for the last

fourteen years, including work on the Wideband Satellite Network, the Butterfly parallel computer and its Chrysalis operating system, the Modular Tactical Gateway project, and network research in time synchronization and multicast protocols. His current focus is on real-time multicast networking technologies.

Commander Dennis K. McBride is Program Manager for Warfighting Simulation, Advanced Systems Technology Office, Advanced Research Projects Agency, Arlington, Virginia. He directs the development of innovative simulation technology to enable manned, force-on-force warfighting preparation and weapon system concept evaluation. Dr. McBride received a Direct Commission in 1979 following completion of the Ph.D. program in Mathematical Learning Theory at The University of Georgia. After completion of Navy Primary Flight Training/Flight Surgeon Program at the Naval Aerospace Medical Institute, he was designated an Aerospace Experimental Psychologist in 1980, and has since completed several tours in Navy RDT&E. His assignments included Principal Investigator for Maintainability Design, Naval Air Development Center; Division Chief for Basic Research in Aviation Man-Machine Interface Design, Naval Aerospace Medical Research Laboratory; Head, Manned Systems Evaluation Lab, providing Simulation T&E Support for F-14D and EA-6B, Pacific Missile Test Center; and Chief Simulation Officer, Manned Flight Simulator - directing T&E Support for TACAIR and various Avionics Programs, Naval Air Test Center. Dr. McBride earned additional post-Doctoral Masters Degrees in Public Administration (Science, Technology & Govt.), and in Systems from the University of Southern California; and completed the Flight Test Engineering Program (University of Tennessee Space Institute) while assigned to NATC. He has over 75 technical publications and reports in systems, aeromedical-human factors, psychological science, and flight test engineering, earned several military decorations, including Navy Achievement, Commendation, and Joint Service Commendation (Desert Storm Impact Award) medals, as well as the L.P. Coombes (Australian Defence Science Organization) Medal. He was chosen by the U. S. Navy as a NASA Astronaut Candidate. His principal flying responsibilities have been in fighter aircraft Test & Evaluation.

SCALEABILITY TOOLS, TECHNIQUES, AND THE DIS ARCHITECTURE

Daniel J. Van Hook, Deborah J. Wilbert, Richard L. Schaffer
Loral Advanced Distributed Simulation
Cambridge, Massachusetts

Walter Milliken
BBN Systems and Technologies
Cambridge, Massachusetts

CDR Dennis K. McBride, PhD.
Advanced Systems Technology Office
Advanced Research Projects Agency

MOTIVATION

The ability of DIS networks to support large scale battle simulations is critical for many anticipated applications of DIS exercises, such as the tri-service Louisiana Maneuvers and ARPA's Advanced Technology Demonstration One. As the size of DIS exercises is scaled up from hundreds to thousands and eventually tens and even hundreds of thousands of simulation entities interacting in simulation exercises, it will become uneconomical and impractical to adequately scale up the supporting communication and computation resources, given the existing broadcast-like DIS communication model. While the DIS standard in principle supports exercises of any scale, limitations will continue to manifest themselves and grow in at least three system components: wide area networks, local area networks, and simulation hosts and their network interfaces. Techniques and approaches are called for to obtain the maximum level of performance at whatever level of communication and computation technology is applied to the problem. Recognizing this need, ARPA is developing a "scaleability" toolset.

SCALEABILITY TOOLSET OVERVIEW

The scaleability toolset is intended to provide a testbed for developing and evaluating solutions to the DIS scaleability problem. The goal is to provide a system that allows "simulation of a simulation" so that alternative approaches may be evaluated at relatively low cost and risk before committing to an architecture or design. The toolset provides a flexible environment for examining the parameters of and interaction between the options available to developers and planners of DIS architectures, exercises, and protocols, including:

- network topologies
- characteristics of links, gateways, etc.
- network algorithms
- exercise and simulation scenarios
- scaling algorithms and techniques for reducing bit and packet per second requirements

With the toolset, a user may develop an exercise scenario and evaluate that scenario in the context of its network resource requirements. The toolset provides a framework to help determine answers to the following important types of questions:

- "To what extent does a proposed scaling algorithm or technique affect network performance?"

- "What approaches to network-level algorithms for congestion control, routing, bandwidth allocation, and delay guarantees are most advantageous for distributed simulation systems?"

- "To what extent will a contemplated exercise scenario utilize available network resources and topology, or conversely, what network resources and topology are required to support a contemplated exercise scenario?"

- "What is the effect of a proposed protocol change on network resources?"

Figure 1 illustrates the functional structure of the scaleability toolset. The major components of the toolset are:

- scenario creation and simulation tool
- scaling algorithm tool
- network simulation and display tool
- scenario viewing and replay tool
- data analysis tools

Scenario Creation and Simulation Tool

The scenario creation and simulation tool, based the ARPA ODIN Semi-Automated Forces (SAF) system [1] is used to develop and simulate exercise scenarios. The toolset user specifies the scenario during an off-line scenario creation phase. The kinds of data needed to specify a scenario include: participating units, unit composition, initial unit positions, distribution of units to network sites, objective(s) for each unit, and specification of the terrain the exercise will be conducted on. The scenario is simulated by the SAF to produce a Scaleability Logger Format (SLF) file that represents in an abstracted form the data packets that would flow between network sites as a result of entity activity and interactions in the exercise. Additional data including timestamps and unit state are added to the output SLF file to provide information needed for other components of the toolset. The SLF format has been carefully designed to optimally trade off storage space, runtime performance, and the data requirements for scaling algorithm development and network simulation. A full discussion of the SAF as modified and used in the scaleability toolset and of the SLF data format and the tradeoffs involved may be found in reference [2].

Scaling Algorithm Tool

The scaling algorithm tool is a flexible testbed for developing and controlling scaling algorithms within the context of the scaleability toolset. It provides facilities for integrating new algorithms into the system according to well defined and documented interfaces. Algorithms are applied to a stream of SLF data (produced by the simulation tool) in order to reduce bit and packet per second rates. Algorithms may be applied in controllable configurations, switched in/out, and parameterized at run time under control of a keyboard interface. The scaling algorithm tool aggregates individual packets into traffic flows that form the input data for and conform to the input interface definition of the network simulation tool.

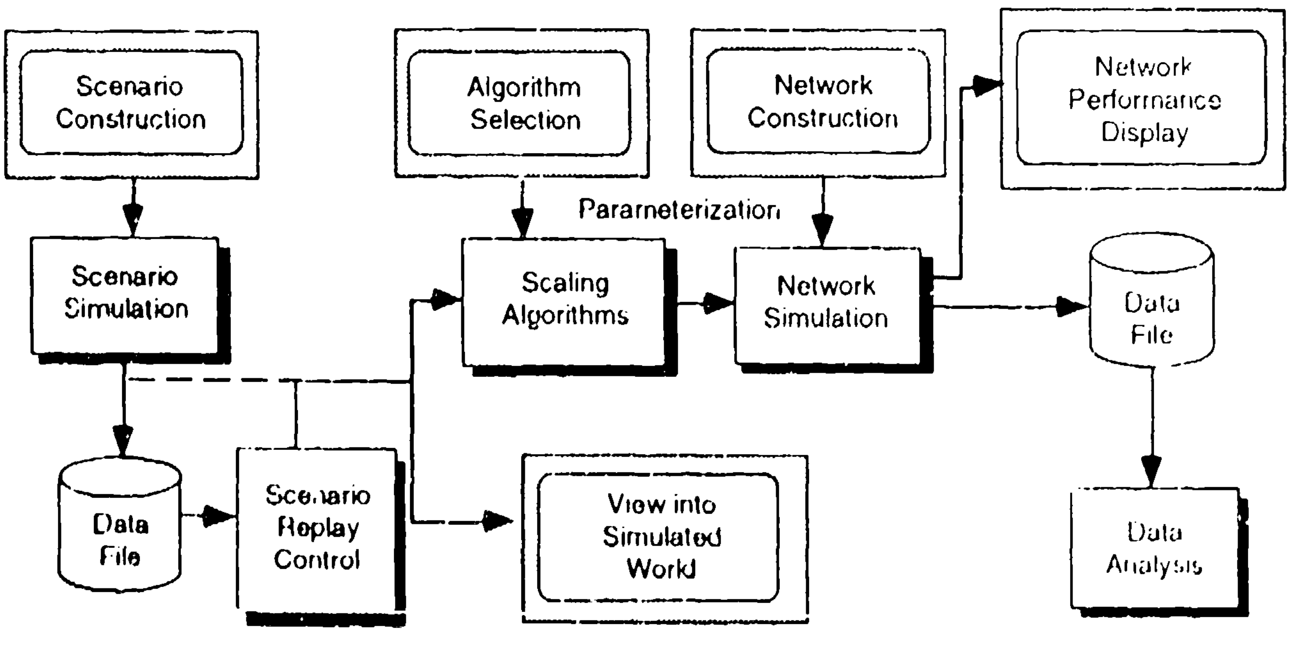


Figure 1 Scaleability Toolset Architecture

Network Simulation and Display Tool

The network simulation and display tool is a flow-based simulator that supports flexible and modular facilities for specifying network topologies and performance characteristics. Networks are modeled as interconnections of network elements such as switches, links, gateways and security devices. The network simulation models aggregated flows through the network as opposed to modeling the transmission of each each individual packet. As such, it produces a medium resolution prediction of network behavior that is consistent with what real physical networks exhibit. A flow-based approach was adopted in order to achieve fast (close to or preferably faster than real-time) processing so as to facilitate a responsive experimentation and evaluation environment. Since multicast capabilities are desirable for several proposed scaling algorithms, the network simulation incorporates support for multicast groups and routing schemes, a feature unavailable in commercially available network simulators. The input interface is a set of standardized messages consisting of timestamped flow and multicast group operations. The network simulator also calculates and gathers extensive performance data, including:

- offered and actual loads for switches and lines
- line transmission and queuing delays
- site LAN outgoing and incoming loads
- delay and drop rates by source site and destination group
- delay histograms of received traffic
- multicast group usage

Run-time performance monitoring indicators include histogram and strip chart displays of delay and loading data and color variation to show utilization levels. In addition, the network simulator includes facilities for outputting performance data to a file for post analysis by commercial data analysis software packages. Both a human-readable ascii/tab-separated format and a more compact self-describing binary format are supported, allowing flexibility in choosing an analysis package. The example analysis described later in this paper made use of BBN/Probe™ [3], which is well suited to flexibly processing and presenting the large volumes of data that are generated. We have made use of the self-describing nature of the binary statistics file format to construct tools that automatically generate analysis functions and scripts to control BBN/Probe processing.

Scenario Viewing and Replay Tool

The scenario viewing and replay tool, based on the CDIN SAF Plan-View Display (PVD) [1], provides a window onto the simulated battlefield. This tool presents the user with a display of the battlefield terrain and the entities participating in the battle, depicted with either unit symbols or as individual entities. By use of this tool, the scaleability toolset user can observe and correlate activity in the simulated world with network performance data reported by the network simulation and display tool. Facilities available to the user include:

- map zoom and scroll
- query an entity for information
- color code forces based on simulation site
- display as units or entities
- terrain analysis: selective terrain feature display, intervisibility, cross
- sections, measurements, coordinate conversions

In addition, a VCR-like control panel is provided for controlling SLF file replay. This control panel supports functions such as pause, play, seek, stop, filename specification, and simulated time display.

The following section illustrates a use of the toolset in detail and the process of analyzing a particular algorithm using a specific network and scenario definition.

SCALEABILITY TOOLSET APPLICATION EXAMPLE

This section provides an example of how the scaleability toolset has been used to investigate a particular scaling algorithm, called grid-based geographic filtering. The methods and results presented here are not an exhaustive characterization of this algorithm but are intended to illustrate how the toolset can be applied to algorithm analysis and by extension used for analysis of networks and simulation scenarios. The series of steps we took to use the toolset for our investigation of the grid algorithm is illustrative of its intended use. Before proceeding to the example, however, it is useful to highlight some of the issues and approaches in grid-based geographic filtering.

Grid-based Geographic Filtering

Geographic filtering is a general technique for reducing the demands a simulation exercise places on the bandwidth, processing, and memory requirements for networks, network interfaces, and simulation hosts. The fundamental concept of geographic filtering is that a simulated entity only needs information about other entities that lie within its region of interest. The region of interest can be thought of as the geographic region of the simulated world that lies within range of the supported sensors, e.g., visual, radar, acoustic, etc. Geographic filtering algorithms arrange to selectively deliver state and event data to simulation hosts based upon the region of interest of locally simulated entities.

One way of implementing the concept of geographic filtering is to associate multicast groups with terrain regions. The multicast routing and addressing capabilities of the underlying network are used to minimize the amount of traffic delivered to each simulator which falls outside its entities' regions of interest. Perhaps the simplest method of assigning terrain regions to multicast groups is to impose a uniform grid system on the simulated terrain and assign a multicast group for each grid cell. This results in each multicast group covering a rectangular solid with a grid cell as its base and infinite height.

Multicast routing and addressing is a developing technology; those networks which support multicasting often impose severe limits on the number of groups and on the rate at which group subscriptions may be changed. While it is appropriate to investigate an algorithm which requires the use of this developing technology, the algorithm should try to optimize its usage and a comprehensive analysis of the algorithm should include an estimate of the requirements for multicast services.

State updates for each entity are sent to the multicast group associated with the cell in which the entity lies. Each simulator joins the multicast groups for the cells that intersect its entities' regions of interest, so that traffic from entities in only those cells are received. As an entity moves across the battlefield, some grid cells will fall out of its region of interest while new grid cells come into range.

Our grid algorithm simulation manages subscriptions to cell multicast groups on a per site basis, whereby an agent notes the state of the locally simulated entities and endeavors to subscribe to the cells overlaid by the union of the regions of interest of the entities simulated at that site. For the purposes of our algorithm analysis, we model an entity's region of interest as an infinite height cylinder with a radius equal to the entity's viewing range. For more optimal cell group subscription, more detailed information about the internal state of each simulated entity than is available to an external entity is required. For this case, it is appropriate for each simulator to determine which multicast groups it should be joining or leaving (rather than being managed externally) since each simulator has the most specific knowledge of the range and scope of its own entities' sensors. Simulators supporting sophisticated implementations of the algorithm could model regions of interest as conic sections for each sensor (by tracking sensor orientation) to reduce the number of unnecessary cell subscriptions. Cell subscription could be further minimized by factoring in range reductions due to terrain or other obstacles (e.g, other entities, weather, chaff).

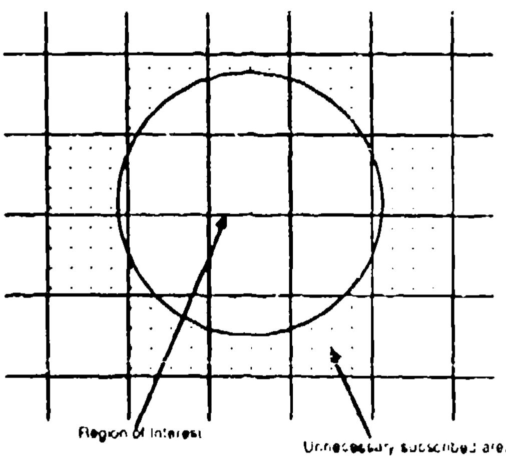


Figure 2 Region of Interest

An important parameter for this algorithm is the cell size. The tradeoff to be considered is that smaller cells result in more efficient filtering, but increase the number of multicast addresses which must be supported by the network and the rate at which the subscriptions to those groups must be changed. As figure 2 shows, a number of cells are only partially overlaid by the region of interest, indicated by the circle. Even though only part of these cells lie within the region of interest, traffic from entities in any part of the cell (including those areas outside the

region of interest) is forwarded. Smaller cell sizes result in less unnecessary terrain being included in the set of cells whose groups are subscribed to, with the potential benefit of receiving less traffic from outside the region of interest. One should also note that in addition to increasing the number of multicast groups which must be supported, smaller cells require each simulator to add and drop multicast groups more rapidly as the entity traverses the terrain.

Enhancements to this basic algorithm may prove to yield substantial benefits. For example, dynamically assigning groups to cells, so that only cells that have entities in them or that lie in the region of interest of an entity actually consume a multicast group. This approach may result in conserving multicast addresses for battle scenarios that have entities widely dispersed across the simulated terrain area.

Another possible extension is that multiple grid systems may allow better control of filtering and more optimal consumption of scarce multicast resources for certain battle scenarios. Yet another possibility is to use a non-uniform grid so that the cells are smaller in regions only where finer grained filtering is required and larger where it is not. Finer grained filtering can be useful in areas heavily populated with entities, so that when another entity's region of interest borders on that area, a large amount of traffic from adjacent entities can be eliminated. Fine-grained filtering could also be used to support particularly heavily loaded network components or simulation hosts by clustering around the regions of interest of the entities served by those components. Optimization to this level would require either carefully coordinated scenario construction and grid cell design or a sophisticated dynamic mechanism for determining the optimal grid assignment on the fly.

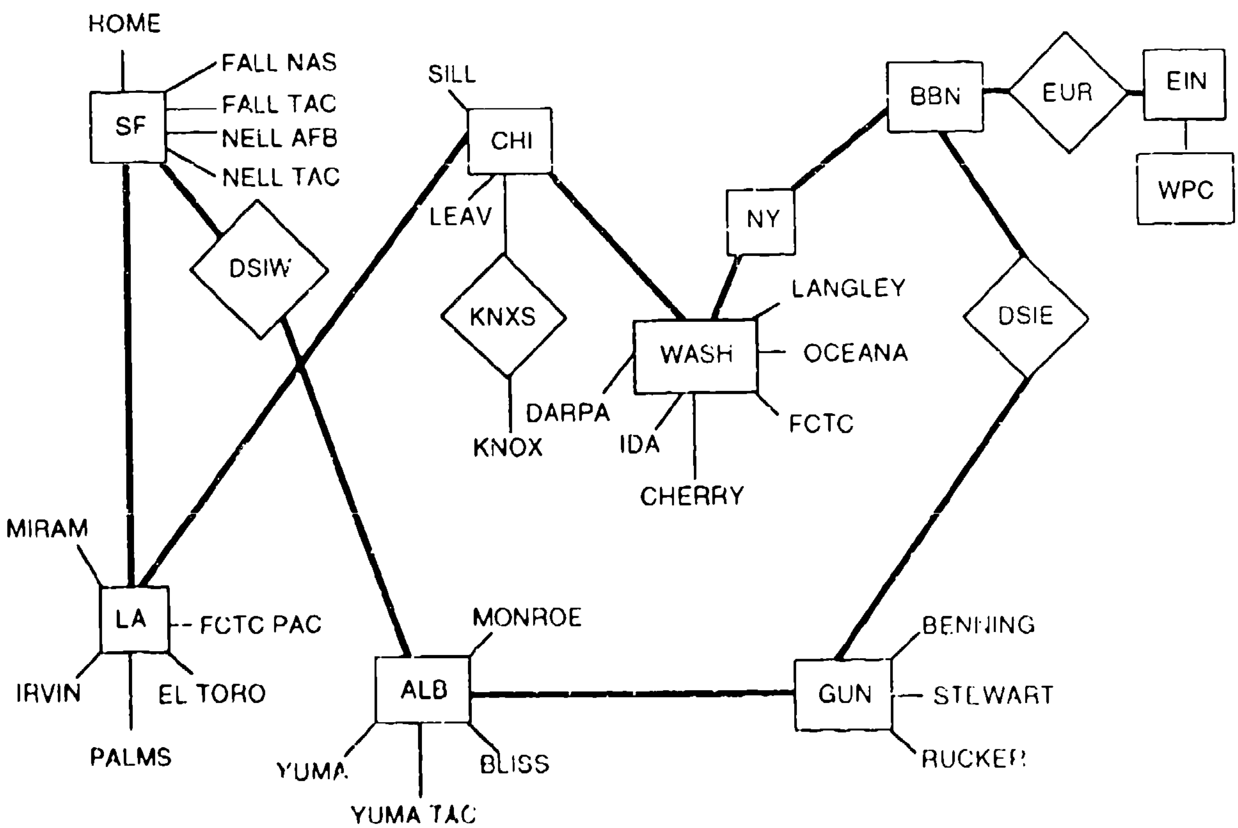


Figure 3 Experimental Network Model

Experimental Parameters and Procedures

Our approach to investigating the effectiveness of the grid-based geographic filtering algorithm consisted of comparing the performance data logged by the network simulator toolset component while the scaling algorithm was in operation (the test cases) with that logged while the scaling algorithm was not in operation (the baseline case). Two test cases were employed: one with a grid cell size of 2.5 by 2.5 kilometers, the second with a grid cell size of 10 by 10 kilometers. The viewing ranges for both test cases were 3.5 kilometers for ground entities and 7 kilometers for air entities (these settings determine the entity region of interest). For both the test and baseline cases, a single battle scenario and network model were employed. For the baseline case (the current DIS communication model), data are broadcast throughout the network, so significantly higher loads were observed.

The network model used in our investigation of the grid-based algorithm is illustrated in figure 3. This hypothetical network is loosely based on the DSI net with the following characteristics and enhancements:

- 27 connected sites
- T1 (1.54 Mbps) tail circuits connecting sites to the backbone
- FDDI LANs at each site
- T3 backbone (45 Mbps)
- switches commensurate in capacity with the T3 backbone

We selected the existing DSI net as the basis of our hypothetical network because it is the only existing network currently being used to host large-scale interactive simulation exercises. We chose to increase the capacity of some network components in order to provide a feasible base for the type of scenario we were interesting in investigating, involving 10,000 or more simulated entities. Figure 3 does not represent an actual planned expansion of the DSI network.

The battle scenario used in the evaluation was developed using the scaleability toolset scenario construction tools with inputs from military subject matter experts so as to insure that it accurately reflected a typical, large scale military exercise. The scenario consisted of five orange divisions (three motorized rifle, two tank) attacking two blue divisions (one tank, one mechanized infantry) across a broad front. Orange and blue aircraft and helicopters carry out missions at intervals throughout the scenario. The orange forces outnumber the blue forces by a factor of approximately three to one, with a total of approximately 10,000 entities represented. The battle takes place within a 50 by 75 kilometer region on simulated Ft. Knox terrain and takes approximately 4 hours of simulated time. The final result is that the orange forces push the blue forces back but the blue forces manage to prevent a complete breakthrough by the orange forces.

The performance measure employed in this study consisted of a measure of the reduction in offered load from the network to the connected sites as a result of applying the grid-based geographic filtering algorithm. This measure may be described as

$$\text{reduction}(n) = \frac{\sum_{m=1}^{M} \text{load}_b(n, m) - \sum_{m=1}^{M} \text{load}_t(n, m)}{\sum_{m=1}^{M} \text{load}_b(n, m)} \qquad (1)$$

where M is the number of sites and $\text{load}_b(n, m)$ and $\text{load}_t(n, m)$ are respectively the loads offered to site m in timestep n for the baseline (no scaling algorithm) and test cases (scaling algorithm in operation). For this measure, the mean reduction over one minute segments spaced at ten minute intervals throughout the battle scenario was gathered and plotted. This measure provides a reasonable high level view of the effectiveness of a scaling algorithm in reducing bandwidth requirements.

Of note in interpreting this performance measure is the fact that the aggregate offered output load will generally contain multiple contributions from each packet that enters the network. Without any traffic reduction algorithm (the baseline case), the network will attempt to deliver every packet to every site. Particularly effective algorithms will reduce (or completely eliminate) the number of sites to which a packet must be delivered.

Experimental Results and Interpretation

Figure 4 graphs the reduction of aggregate output load reduction, described in the previous section, achieved by the grid algorithm as a function of simulated time. Results are shown for two grid cell sizes: 2.5 by 2.5 kilometers, and 10 by 10 kilometers. The

more effective filtering action of the smaller cell size is evident from the figure. As described previously, this effect can be attributed to the reduction in area that is part of the cell multicast groups subscribed to that lies outside the circular region of interest with a consequent reduction is unnecessary received traffic. Of note is the drop off of effectiveness as the battle scenario progresses. This drop can be attributed to intermixing of the simulated military units which occurs later in the battle scenario - more entities from different sites come into contact (are positioned within each other's regions of interest) later in the exercise, requiring more traffic to be forwarded.

Two cautions should be noted when interpreting these results. First, while this measure of effectiveness yields useful insight into how well a scaling algorithm will perform in the aggregate, individual network elements may still be overloaded even when this measure predicts outstanding performance. This effect was in fact observed while conducting these experiments. Therefore, it is important that any full characterization of an algorithm requires that additional and diverse performance measures and approaches be applied. Relying on only a small set of characterizations would be unwise. We are currently investigating other ways of calculating measures of effectiveness for algorithms, which include considerations of network congestion, dropped packets, delay and individual network element performance. Second, we believe that the network topology and battle scenario may strongly influence the results obtained for any particular measure of effectiveness. Therefore, it is important that any full characterization of an algorithm requires experience with diverse scenarios and networks in order to determine their impact on scaling algorithm performance.

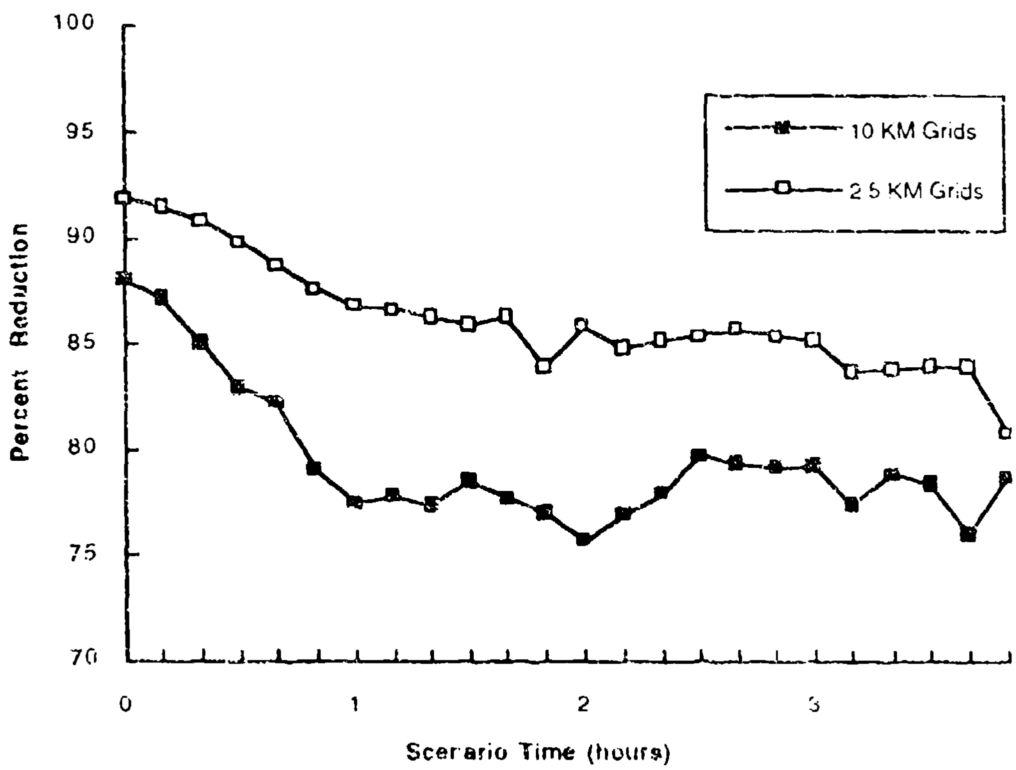


Figure 4 Reduction in Aggregate Offered Output Load

CONCLUSIONS

A scaleability toolset is being developed for the purpose of providing a testbed for developing solutions to the DIS scaling problem. This toolset enables development and evaluation of DIS exercise scenarios, scaling and network algorithms and techniques, and network topologies in a simulation environment, i.e., a simulation of a simulation. Not only can the toolset be applied to algorithm analysis, as illustrated by the example given in this paper, but it may also be applied to battle scenario development and network development.

As our analysis shows, grid-based geographic filtering algorithms that use multicasting can reduce network traffic dramatically over broadcast schemes. Therefore, it is fruitful to pursue further investigation of this promising family of algorithms for solving the scaleability problem and also to consider enhancing currently available multicast facilities. We are currently pursuing the analysis of enhancements to the simple single-grid algorithm. In addition, we are also considering other forms of geographic filtering that do not make use of grid-based techniques as described in this paper and also other families of algorithms which could be coupled with geographic filtering algorithms to form a comprehensive approach to solving the scaling problem.

Additional useful measures for characterizing scaling algorithm performance in terms of delay reduction, dropped packets, congestion characteristics, and individual link and switch loading have been developed or are under development. Our experience with these new approaches will be reported in the future.

More research, both in terms of possible algorithms and ways of measuring their effectiveness, will provide a good foundation for selecting appropriate traffic reduction algorithms. Good algorithm selection will be needed for DIS exercises to meet the goal of very large scale tri-service training on networks of the foreseeable future.

ACKNOWLEDGEMENTS

The development of the scaleability toolset is being sponsored by the Advanced Research Projects Agency, Department of Defense and monitored by the Topographic Engineering Center under contract DACA 76-91-C-0006. CDR Dennis K. McBride, USN, is the ARPA Program Manager and has provided the guidance. Debbie Deutsch, Josh Seeger, and Shawn Smith of BBN Systems and Technologies and Barbara Perry, Maureen Saffi and Rob Vrablik of Loral Advanced Distributed Simulation have all made significant contributions, as has Mr. David A. Fusco of the Naval Command, Control and Ocean Surveillance Center, RDT&E Division (NRaD), San Diego, CA.

REFERENCES

1. Saffi, M. "The OBG/SAF Interface User Guide," version 4.2.0, Bolt, Beranek, and Newman, Inc., August 14, 1992.

2. Vrablik, R. and Wilbert, D. "The Use of Semi-Automated Forces to Simulate a 10,000 Entity Exercise," presented at the Third Computer Generated Forces and Behavioral Conference, March, 1993, Orlando, Florida.

3. "BBN/Probe Command Reference Manual", BBN Systems and Technologies, 1992

COST-REDUCTION FROM SIMULATOR DATA BASE REUSE: FEASIBILITY OF REFORMATTING A-6/F-14 SIMULATOR DATA BASES FOR THE DOD STANDARD SIMULATOR DATA BASE PROJECT 2851

Douglas E. Dillard, Budimir Zvolanek, and F. Donald Eckelmann Jr.
McDonnell Douglas Training Systems
McDonnell Douglas Corporation
St. Louis, Missouri

ABSTRACT

Current estimates show that approximately one-thousand image generators are now in use for a variety of simulation applications. The cost of developing new data bases for these and future image generators is tremendous. One way to reduce up-front costs is to reuse existing simulator data bases rather than generate databases from scratch.

This paper describes the results of investigations undertaken by McDonnell Douglas Training Systems into the feasibility of reformatting three data bases into the Project 2851 Standard Simulator Data Base (SSDB) Interchange Format (SIF). These data bases, originally developed for the suite of A-6/E S/WIP and F-14D trainers, support visual, infrared, and radar simulations over large areas of the East and West coasts of the United States. They would be valuable and cost-effective additions to the Project 2851 repositories. We evaluated the following aspects of the A-6E S/WIP and F-14D data bases:

Levels of Detail	Feature Representations	Terrain Representations
Model Formats	Texture Representation	Infrared Attributes

We conclude that it is feasible to reformat terrain elevations, cultural features, models, and textures, given certain conditions on spatial resolution, thermal attribution, and texture representation. The results uncovered a need for minor changes to the SIF/HDI standard format.

ABOUT THE AUTHORS

Doug Dillard is a Lead Engineer in Technology Development and Integration at McDonnell Douglas Training Systems (MDTS). Mr. Dillard has concentrated his career on the design of visual and sensor simulators and data bases for computer image generators used in aircrew training systems. He worked on research contracts for Project 2851 and on internal research on data bases at MDTS. Mr. Dillard received a B.S. degree from St. Louis University in St. Louis, Missouri, and an M.S. degree from the University of Missouri at Rolla, Missouri.

Budimir Zvolanek is a Technical Specialist in Technology Development and Integration at MDTS. Mr. Zvolanek has concentrated his career on the application and development of electronic imaging systems, sensor image processing, and database development. He is the principal investigator in image simulation and database technology development and was a program manager for the DoD Standard Simulator Data Base Project 2851 contracts at MDTS. Mr. Zvolanek received his M.S.E.E. and B.S.E.E. degrees from Washington University in St. Louis, Missouri.

Don Eckelmann is a Senior Engineer in Design at MDTS. Mr. Eckelmann has devoted his career to the application of computer graphics to flight simulators. He has worked on the design and integration of large-scale flight simulators, computer image generation systems, and data bases. He received his B.A. from Colgate University.

COST-REDUCTION FROM SIMULATOR DATA BASE REUSE: FEASIBILITY OF REFORMATTING A-6/F-14 SIMULATOR DATA BASES FOR THE DOD STANDARD SIMULATOR DATA BASE PROJECT 2851

Douglas E. Dillard
Budimir Zvolanek
F. Donald Eckelmann Jr.
McDonnell Douglas Training Systems
McDonnell Douglas Corporation
St. Louis, Missouri

INTRODUCTION

A rough count of currently fielded flight simulators shows that there are over one-thousand computer image generator (CIG) systems used for a variety of aircrew training applications[1,2], with over one-half devoted to military applications. Although the data bases for many of these systems represent the same geographic areas, many were independently generated by CIG manufacturers. For example, at least four CIG manufacturers are represented at North Island NAS, and each manufacturer generated its own North Island data base. Such ab- initio generation of new CIG data bases does not capitalize on efforts put into development of prior data bases which depict the same territory.

In addition, current requirements for correlation are driving upward the already high cost of developing data bases for computer image generators:

- Correlation between Out-The-Window (OTW) visual systems and multiple sensor systems, such as infrared (IR) and radar.
- Correlation between multiple simulators with varying degrees of capability in their image generators.
- Correlation between image generator data bases, operator displays, and real time systems.

One way to reduce up-front costs is for the Department of Defense (DoD) to reuse existing simulator data bases to populate the repositories of the Standard Simulator Data Base Project 2851 (P2851)

AN OPPORTUNITY FOR REFORMATTING

McDonnell Douglas Training Systems (MDTS) has been under contract to develop common, sensor, and visual data bases for the A-6E System Weapons Integration Program (A-6) and F-14D (F-14) aircraft simulation systems. These data bases represent years of development effort and would be a valuable addition to the central repositories of the P2851 Standard Simulator Data Base (SSDB)[3] since they:

- Cover over 500,000 square miles of the United States
- Contain data for OTW visual and infrared and radar sensor simulations
- Contain enhanced areas for targets, airfields, radar, and visual areas of interest.

We describe solutions which are compatible with the SIF/HDI standard. Where appropriate, we recommend ways to improve the standard, preserving expended effort, and extending the SIF/HDI standard.

DESCRIPTION OF THE SOURCE DATA BASES

The A-6 and F-14 data bases represent three gaming areas within the United States, each referred to by its primary airfield. These data bases, illustrated in Figures 1 through 3, support OTW visual, infrared, and radar CIGs. The **WHIDBEY** data base, for the A-6 program, represents approximately 224,000 square nautical miles, mostly over Washington and Oregon. The **MIRAMAR** data base for the F-14D program, represents approximately 72,000 square nautical miles of the southwestern United States. The **OCEANA** data base is shared by both programs and represents approximately 224,000 square nautical miles of the eastern United States.

Common Data Base Components

The Evans and Sutherland Corporation generated a common data base for each of the three gaming areas. Each data base is a collection of

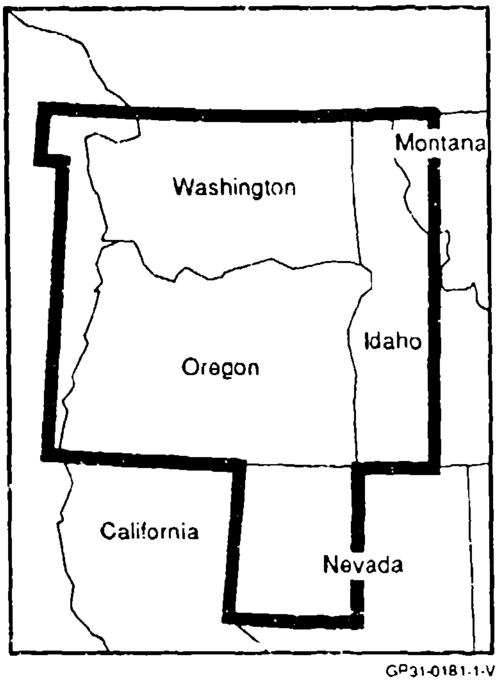


Figure 1. The A-6 Whidbey Data Base Gaming Area

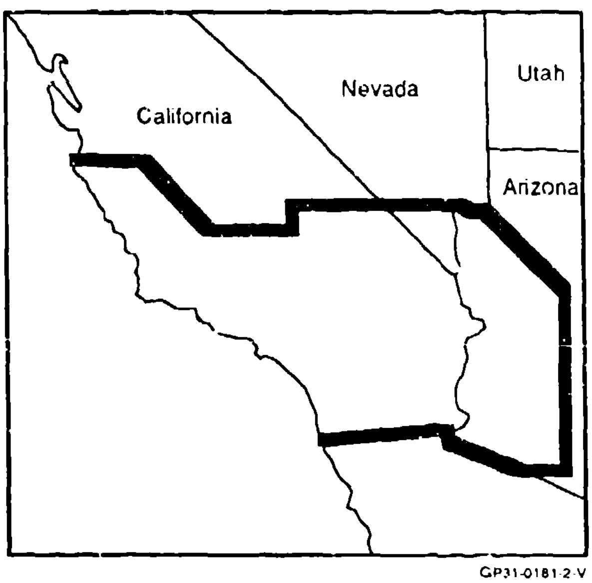


Figure 2. The F-14 Miramar Data Base Gaming Area

component files which are in a proprietary format. The files contain data for visual and the Forward Looking Infrared (FLIR), and radar sensors. Depending on how they are compiled, these files may produce visual data bases, infrared data bases, or radar data bases. A common data base contains the following elements:

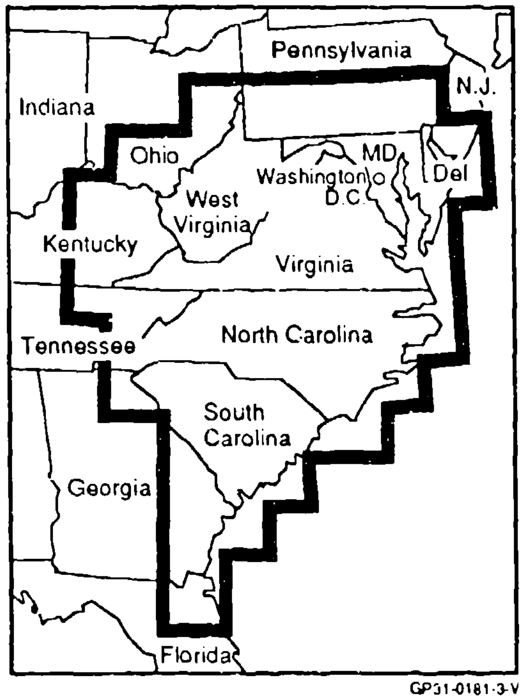


Figure 3. The A-6/F-14 Oceana Data Base Gaming Area

Objects – An object is a set of scene elements, polygons, and light strings, with fixed visual priority.

Models – Models are three-dimensional representations of features constructed from polygons and light points. They enhance the terrain and represent real-world and threat-world features. Models can be stationary or dynamic.

Texture – Visual texture provides altitude and velocity cues required for high-speed, low-level flight while minimizing the number of terrain and culture polygons. The visual texture has been specifically designed to support low-level flight. In both visual and radar simulation, texture also adds realism to models of buildings, ships, and aircraft by providing surfaces which appear real. The texture patterns, or texture maps, are constructed either algorithmically or from photographs.

Plane Features – Plane features are non-visible features which determine the visual priority among groups of visible features.

Terrain Representation – Two terrain representations, the result of several years of automated and manual modelling effort, are used to form the run-time data bases. The first is a 100-meter, gridded representation used to create radar run-time data bases.

The second is a coarse polygonalization of the earth's surface for visual and IR simulation. The polygonalized terrain has been forced to exactly match the gridded terrain at designated points to insure point correlation. While the automated parts of the processes could be repeated by starting with the original data and applying similar algorithms, the results of the manual efforts involved in supervising, evaluating, and editing will be lost if <u>both</u> representations are not retained.

Benchmark Features – Since the visual and radar run-time data bases use data of differing real-world accuracy for terrain representations, the terrain data may differ substantially. Benchmark features are used where a high degree of correlation between various sensors is required. The features specify boundaries within which terrain elevations must exactly correspond in both the visual and radar simulations. The tools which generate the radar run-time data base use benchmarks to ensure that the visual and radar data bases are correlated exactly in such areas by forcing the terrain within the benchmark areas to be equivalent in radar, visual, and FLIR data bases.

Run-Time Data Bases

An Evans & Sutherland ESIG-500 image generator processes and displays the contents of the visual and FLIR run-time data bases. The image generator uses its inputs and a set of data structures which model the run-time data base to determine the items to be displayed, their visual priority with respect to one another, and other image characteristics.

The MDTS Advanced Radar Image Generation System (ARIGS) radar simulator processes and displays the radar run-time data base to produce a simulated radar display. A run-time radar data base is a gridded data set which models the gaming area at a number of different resolutions. The radar data base formatter partitions the data base into square tiles, 256 posts on a side, with the posts spaced evenly in flat earth coordinates. Each post has data fields which specify elevation, reflectance, dispersion, directivity, and other attributes.

PROJECT 2851

Project 2851 is a Research and Development program chartered by the Joint Technical Coordinating Group for Training Systems and Devices. The objectives of the program are to reduce the costs of generating data bases for DoD training systems, increase data base performance, and address problems of correlation within and between these systems. Cost reductions will be achieved by eliminating duplicate data base generation and redundant software development.

Project 2851 will result in a DoD Simulator Data Base Facility which will obtain standard Defense Mapping Agency products, externally generated simulation data bases, and other source material; archive and manage this data; and provide tailored data base products to DoD training simulators.

The SSDB Interchange Format (SIF)[4] will serve as the primary vehicle for sharing externally developed digital data bases across programs and services. Each time a new simulator program requires a data base, it will be able to access all relevant existing data bases maintained by Project 2851. Conversely, when a program has enhanced a data base, it will be able to send it to Project 2851 and make it available for other projects.

The SIF standard encompasses two formats, SIF for High- Detail Input/Output (SIF/HDI) and SIF for Distributed Processing (SIF/DP). The SIF/HDI format is the most appropriate format into which to place the A-6 and F-14 data bases, since their reuse represents an exchange between an external simulator site and P2851.

The second standard, SIF/DP, intended for distributed processing and supplementing the P2851 resources, allows data base exchanges in a format which is nearly identical to the internal P2851 format. It supplements the P2851 computational resources and facilities.

TRANSFORMATION OF THE DATA BASES INTO SIF/HDI FORMAT

The requirements of radar, visual, and infrared displays force the A-6/F-14 common data base to meet the conflicting needs of two different imaging systems:

- High resolution, low display rate radar system.
- Low resolution, high display rate visual and IR systems.

Levels of Detail

The conflicting requirements force two Levels Of Detail (LODs) for features and terrain. Since the

radar system has a low display rate, it can tolerate a high density data base which provides a very detailed representation of terrain and culture. Because the visual and IR systems have high display rates, which limit processing time, they require much less detail in their representation of the world.

Recommendation – The SIF representations of the common data bases should have two LODs. The two levels will meet the needs of both the high-resolution radar and the low-resolution visual communities and will preserve the effort already expended to meet these needs under the A-6 and F-14 projects.

Transfer of Terrain Representations

We have examined how to place data in the files, records, and fields of the National Imagery Transmission Format[5] (NITF), which the SIF Standard mandates for gridded terrain representations. There are no insurmountable problems in using this format to store gridded elevation data.

Recommendation – Because a significant amount of effort went into the generation of the two terrain representations, both representations should be retained: Use the 100-meter-gridded-elevation terrain for the best level of detail; use the polygonized terrain as areal features for the lesser level of detail. This combination provides the maximum possible value from the common data base, and meets the needs of a diverse user community.

Transfer of Culture/Feature Data

Each feature in the common data base is marked for applicability to visual, infrared (IR), and radar image generators. The Evans and Sutherland compilers use only those features marked for their targeted simulation. The visual and infrared compilers use only features marked for their use. The radar compiler selects the features marked for radar (often more detailed versions of the visual and IR features) and identifies the models which replace point features.

SIF/HDI Features – The SIF allows areal, linear, point, point-light, and point-lightstring features. It also allows two- and three- dimensional models both as replacements for features and as stand-alone features. Figure 4 shows a simplified model and feature placement for the SIF/HDI and a simplified diagram of the method by which SIF specifies a model which does not replace a feature.

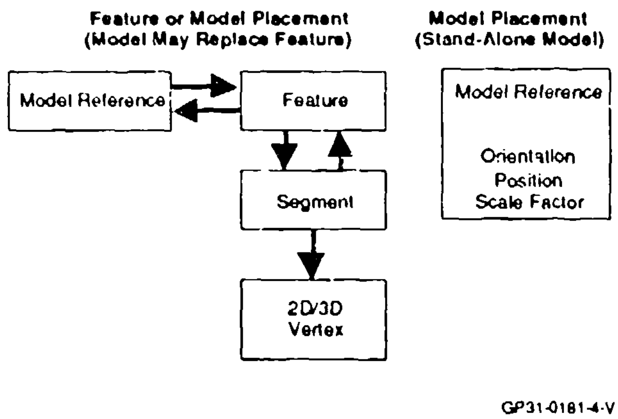


Figure 4. Simplified Model and Feature Placement for SIF/HDI Format

It is feasible to reformat features from the A-6/F-14 common data base into the SIF/HDI format. The section on processing describes the software required to transfer features to SIF.

Transfer of Models

The static models in the common data base are three dimensional (3-D) polygonal representations of point features. They are high LOD representations of the models used for the visual run-time data bases. The dynamic and relocatable models are 3-D polygonal representations of aircraft, surface craft, missiles, and other moving and relocatable objects. They are constructed in the same way as are the static models, and are in the same format as static models.

We have found the transfer of models possible, but some amount of processing software must be created to accomplish the transfer. Since dynamic and relocatable models are constructed the same way as the static models, these models may also be transferred. Some missile models may require an axis exchange to place the Z axis along the direction of motion, to conform with the SIF model building standards.

Transfer of Texture

The common data base employs a generic texture type, called a modulation map, for both the OTW visual and FLIR displays. Modulation maps are texture element (texel) arrays in which the value of each texel determines a mix of two color values, a primary and a secondary, to form a resultant texture

color. The value associated with each texel is a modulation multiplier, a coefficient in the combination of the two texture colors:

$$\text{Resultant Color Value} = (1/255) * \left[\begin{array}{c} \text{Texel\_Value} * \text{Primary\_Color\_Value} \\ + \\ (255 - \text{Texel\_Value}) * \\ \text{Secondary\_Color\_Value} \end{array} \right]$$

GP31-0181-30-V

For example, if a given texel has a value of 200, then the color for that texel is 200/255 times the primary color value plus (255-200)/255 times the value of the secondary color. In this way, up to 256 different colors could be present in a given texture pattern while using only two entries in a color LUT. Modulation maps are widely used throughout the common data base for terrain and cultural features such as grasslands, deserts, agricultural fields, broken clouds, urban areas, and airfield features.

General Use of Modulation Maps – Modulation maps are used by a wide variety of CIGs. However, in each installation of an image generator and display system, the patterns must be tuned to the characteristics of the actual system. For example, the primary and secondary color RGB values should be adjusted to accommodate the degree of contrast provided by a display system.

SIF Options for Texture – The SIF/HDI format has provisions for generic texture. According to the SIF standard[4], the generic texture consists of "non-geospecific images for ... geographic areas...".

The NITF file types specified by SIF/HDI can easily contain texture data from the common data base. Modulation Maps can be stored by either of the two methods provided by SIF/HDI:

ACTUAL VALUES:

"the actual band value(s) at that texel position (e.g., the red, green, and blue intensity values)"

INDEX/LUT:

"an index into a look-up-table (LUT) which is defined in the Image Sub- Header File."

Recommendations – The Index/LUT method should be use to store modulation texture maps as SIF/HDI generic texture maps Because it is most similar to a modulation map, it will occupy about two thirds less storage space than the other method, and it will be easier to convert back to the original modulation map structure. This method gives the user the opportunity to tune such a texture by modifying only two colors at a time, versus 256 or more.

The SIF standard be extended to allow storage of modulation patterns. Storage of these patterns would be fairly simple and would alleviate operational problems associated with re-deriving modulation maps from the Index/LUT maps.

Radar Texture

The SIF standard does not allow the inclusion of radar-specific texture patterns which represent features in the radar run-time data base. The SIF will hold neither the texture for the run-time data base nor will it hold a gridded representation of the data from which the patterns for the run-time data base can be derived.

The Radar Data Base Formatter, which produces the run-time data base, uses feature attributes to transform features into gridded representations which are conceptually very similar to SIF/HDI texture representations. The representations are arrays of data posts, evenly spaced in ground plane (X,Y) coordinates, with each data post representing the area surrounding it. Each data post contains a height representation and hardware-dependent codes for vertical/horizontal, reflectance, directivity, and dispersion. The SIF has no provisions for generic height, material, or use patterns.

Recommendations – The standard should be extended:

- The SMC/FDC pattern should also represent generic features.
- The gridded height representation should be extended to represent non- specific textured heights above terrain.

Coding of Infrared Attributes

The SIF cannot represent all the types of materials and material compositions provided in the common database. In the common database, an Extended Material Code describes the material composition of each model polygon. This code categorizes a material and its properties to a greater extent than do existing DFAD and Project 2851 codes. Examples

of the material codes from the A-6 and F-14 Data Base Design documents[6,7,8] are:

Code	Material	Material Characteristics
10	Aluminum	Dull, Thin
15	Aluminum	Dull, Thin
20	Aluminum	Polished, Thin
25	Aluminum	Polished, Thick
190	Leaves	Live

GP31-0181-31-V

Since there are many more material codes than the 14 valid Surface Material Category (SMC) Codes, the SIF format cannot represent the range of materials presented in the common database.

Recommendations – There are two choices for thermal attributes:

- If thermal models are available, derive the SIF attributes. Use either the existing Evans and Sutherland thermal model or another model.
- If no derived attribute can be made available, do not populate the thermal attribute fields, since the SIF does not provide for a sufficient representation of materials.

In addition, the SIF standard should be extended to better specify thermal attributes:

- Allow more than 14 material codes by defining material subtype fields.
- Explicitly define related FACS fields (living/dead, thick/thin, weathered, polished/rough, and others).
- Explicitly define FACS fields for thermal conductivity and specific heat.

A companion paper contains additional recommendations[9].

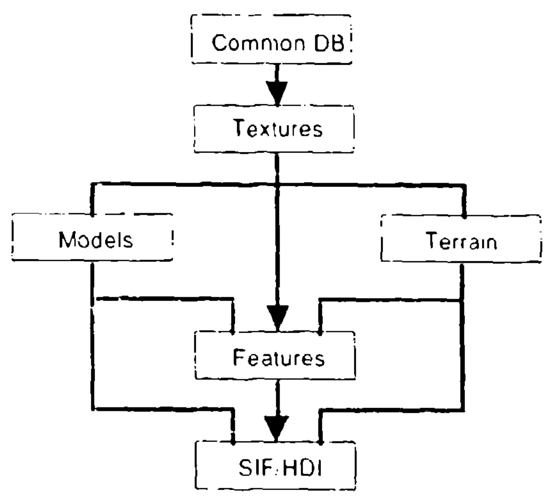
THE PROCESS OF REFORMATTING

The conversion process from the A-6/F-14 common data base to the SIF/HDI format is shown in Figure 5. The process flow is depicted in terms of the four major components in the process and their various conversion order dependencies:

- Textures
- Models
- Terrain
- Features

Since textures are independent of the other three A-6/F-14 common data base components, and are used by all the other components, they are first in the conversion process. Although models and terrain are mutually independent, both are dependent on textures which precede them in the process flow. Terrain is then the basis for features which are placed upon it. Not only do features derive their altitude from the terrain on which they are placed, but they are dependent on all three of the other common data base components in this process, and thus are the last item in the flow. The steps required to reformat any data base element are:

- Convert positions into latitude, longitude, and elevation
- Partition along lines of latitude and longitude
- Map common data base features into SIF/HDI features
- Provide model placements and replacements
- Place data into SIF/HDI files



GP31-0181-5-V

Figure 5. Reformatting the A-6/F-14 Common Data Base Into SIF/HDI Format

ISSUES

The source data bases have several limitations which may require additional work for other applications:

- The high resolution terrain data have small discontinuities. Although these discontinuities are within DMA quality standards, they may create artifacts in high resolution radar simulations.

- There is a restricted set of models for point features
- There is a restricted set of heights for models
- Enhanced areas have sharply defined edges. (postage stamp effect).

CONCLUSIONS

Our investigations have resulted in the following conclusions:

1. It is feasible to reformat these elements from the A-6/F-14 common data bases.

    - Terrain elevations
    - Cultural features
    - Models
    - Textures

2. It is desirable to reformat these data bases since the simulation community will gain use of very large data bases containing many enhancements, and customers will save on database generation costs.

3. The process of reuse can be assisted by some minor changes to the SIF/HDI standard.

ACKNOWLEDGMENTS

The authors thank Gene Clayton, P2851 Program Manager at PRC, and his team for supporting these efforts.

REFERENCES

1. Warwick, Graham. "World Airline Simulator Status", Flight International, 8 - 14 April, 1992.

2. Warwick. Graham. "Simulated Action", Flight International, 28 October - 3 November, 1992.

3. Project 2851. Data Base Design Document (DBDD), Standard Simulator Data Base (SSDB). McLean, Virginia, Planning Research Corporation, for AFSC/ASD/YT, 1990.

4. Project 2851. Military Standard (draft) Standard Simulator Data Base (SSDB) Interchange Format (SIF) Design Standard. PRC, McLean, Virginia, 1993.

5. Defense Intelligence Agency. National Imagery Transmission Format (NITF), version 1.1. Arlington, Virginia: Defense Intelligence Agency, DIA/DM-1A, 1989.

6. Evans and Sutherland Simulation Division. Database Design Document, McDonnell Douglas Training Systems: Miramar. Evans & Sutherland, 1992.

7. Evans and Sutherland Simulation Division. Database Design Document, McDonnell Douglas Training Systems: Oceana. Evans & Sutherland, 1992.

8. Evans and Sutherland Simulation Division. Database Design Document, McDonnell Douglas Training Systems. Whidbey. Evans & Sutherland, 1992.

9. E. Bauman, B. Zvolanek, and D. Dillard. "Infrared Attributes for Project 2851 Standard Simulator Data Bases," Proceedings of the 15th I/ITSEC Conference, November 1993.

DEALING WITH A VARIETY OF RESOURCES IN DIS IMPLEMENTATIONS

John O'Reilly
Reflectone, Inc.
Tampa, Florida

ABSTRACT

Today's simulators involve many varied computational systems. Interoperating these devices is a key strategy for extending the life of current simulators. The Distributed Interactive Simulation (DIS) protocols solve the maze of interoperability.

These statements are well-known to the simulation industry, but the stark reality of the situation is that varied systems produce varied implementations of DIS. The full gamut of DIS involves the varied processors, varied operating systems, varied programming languages and structures, varied interface hardware, varied coordinate system implementations, and varied data base formats. Realizing these six areas simultaneously is a particularly demanding chore. This paper attempts to show how one implementor went about producing code for DIS, seeking to provide reusable code in the process. The lessons learned from this venture are discussed.

Using a PC-based radar simulation system as the baseline, the paper discusses the research and development of DIS in this varied environment. Although a radar appears to be a "receive-only" entity on a DIS network, in order to locally test such a system, test vectors (or PDUs in the DIS parlance) must be generated. Thus, the baseline requires some way to construct test vectors, such as through semi-automated forces (SAFOR) or Computer Generated Forces (CGF) generators. A limited CGF for the required purposes is described in the paper.

Other steps in the implementation include actually producing, transmitting, receiving, and displaying the CGF state vectors. Production involves coordinate conversion schemes, PDU receive and transmit functions become as dissimilar as their associated processors, and display techniques require limitations in the scope of what can be displayed. So, the paper surveys network I/O techniques and selects the correct one for the radar simulation. The last stage (displaying) requires a filter of the vectors since all processors (and especially the PC in question) have limits in terms of computing power.

Intermediate steps in the full implementation of a DIS system involve determination of correct protocols sent from the CGF and the use of terrain and feature data bases. Both of these areas are also discussed including the fields of network analyzers, DMA maps, and Project 2851 SIF.

The paper points out that, although realizing a DIS interoperation can be straightforwardly done, care must be taken to understand that there is more to the "varied" problem than just the obvious processor incompatibilities.

BIOGRAPHICAL SKETCH

John has worked in the simulation industry for 25 years beginning in the CAE world of circuit simulation using SCEPTRE and SPICE. He has been in the employ of Reflectone for ten years working variously as a controls engineer, computer systems manager, acoustic simulation engineer, radar system engineer, and R&D investigator. Research interests and experience include artificial intelligence, radar simulations, data base design, and DIS.

John holds a BSEE and MSEE from the University of South Florida in Tampa and has done post-graduate work in multi-disciplinary simulation.

DEALING WITH A VARIETY OF RESOURCES IN DIS IMPLEMENTATIONS

John O'Reilly
Reflectone, Inc.
Tampa, Florida

INTRODUCTION

Interoperability is both the blessing and the curse of the Distributed Interactive Simulation (DIS) implementor. Allowing us, by definition, to communicate seamlessly between networkable computers using pseudo-real-time protocols provides many coding advantages:

- Drop-in code - We can develop code once for many platforms (although this is not necessarily accurate, as we will see),
- A functional separation of decomposition is readily available,
- Information hiding design methodologies fit in well from processor-to-processor, and
- Standard protocols are easier to debug, assuming a network analyzer is available.
- A system with the DIS interoperability capability is an "easy sell" in the competitive simulation industry.

But DIS has its disadvantages, too. We will discuss these problems areas with an example.

The 1992 I/ITSEC conference's state-of-the-art theme involved a demonstration of DIS. Over twenty corporations were involved in defining, discussing, implementing, and demonstrating the use of DIS protocols during the three-day affair. This was the first major exhibition of DIS's capabilities in a public forum - a vast undertaking in its own right - and the first undertaking by this researcher of any coordinated network simulation. Reflectone chose to put a "listen-only" Digital Radar Landmass Simulation (DRLMS) in the demonstration network. Although this listener was only expected to receive network packets, the basic process of appearing on the network was found to be equivalent to performing a full-scale interoperability simulation. This concept will become more apparent as we discuss the implementation and its attendant problem resolutions.

In order to place a radar simulation system on the I/ITSEC network, a full set of DIS Protocol Data Unit (PDU) software was required. Entity State, Fire, Detonation, and Collision PDUs would have to be sensed from the network, decoded accurately, and converted to the pre-existing radar simulation's coordinate system. The radar simulation was hosted on a PC-compatible machine running the standard one megabyte operating system. No prior experience on this use of networking protocols, especially UDP/IP (User Datagram Protocol/Internet Protocol, as defined by the DIS Standards), was available for the host computer. As an additional sidelight, the I/ITSEC demonstration made the Project 2851 Simulation System Database Interchange Format (SIF) format database the standard. This aspect added more complexity to the DIS Interoperability Demonstration problem, although SIF is completely distinct from DIS in principle.

The scope of the DIS standards were somewhat reduced by popular opinion (e.g., limited set of PDUs allowed on-line, simplified coordinate system, and broadcast-only packets employed), but the mainstays of DIS were left in place (i.e., full entity state PDUs, UDP/IP protocols, dead-reckoning algorithms, geocentric coordinate system).

Having inroduced the scope of the problem (DIS Interoperability Demostration using a DRLMS), the remainder of this report will discuss the implementation of the DIS and DRLMS requirements.

VARIED RESOURCES ARE NEEDED TO MEET REQUIREMENTS

The Institute for Simulation and Training (IST, the organization tasked with both putting together the DIS Standards and the 1992 I/ITSEC Interoperability Demonstration) realized that some means of testing DIS systems was necessary. Thus, IST generated a test procedures document which all participants in the demonstration were required to pass before being certified to interoperate on the I/ITSEC network. The document also provided processes by which IST would verify passage of the test procedures. However, this document and its processes were not available early enough in the development cycle to provide any testing methodology by the participants. Thus was born the "FLY" program.

FLY is a semi-automated forces generator (SAFOR) specifically designed to produce accurate and (for testing purposes) repeatable DIS PDUs. It was decided that it would be nice to host the FLY routine on a processor unlike the target radar processor, so FLY was generically

designed and implemented in C to operate on both a VAX system and a Motorola UNIX system. The choice of the processors was foremostly made because of availability, but secondarily made to locate and understand how DIS should and could be implemented for true interoperability.

The widely varying aspects of the VAX with the VMS operating system and the Motorola Delta box with its UNIX operating system injected some unexpected difficulties. The first concern was raised by the speed of all of the processors. Relative CPU horsepower is certainly a concern for any networked design due to throughput considerations – Is the processor capable of transmitting and receiving and decoding DIS PDUs in a timely fashion? It turned out that the PC was probably the most efficient in this aspect, because of its single user operating system as opposed to the multi-user VMS and UNIX environments. In order for VMS and UNIX to be as sophisticated as they are, they must provide packet trapping mechanisms well above those of the DOS-based PC operating system. The major concern here was thus to reduce the effect of the operating system on the application's design limitations. The FLY and Radar simulation applications should implement the dead reckoning and coordinate transformation algorithms as efficiently as possible, cutting corners as much as possible. Assumptions of limitations will be discussed more in the next section.

The varying host processors necessary to implement any DIS design also have internal hardware variances. The hardware implementation of floating point numbers is a distinct problem area. The DIS standard quotes the IEEE Standard for Binary Floating Point Arithmetic (IEEE 754-1985) as the required standard. This standard is implemented in many processors directly (in this case, the Motorola 680X0 and Intel 80X86 processors) but is not implemented in others (specifically, the VAX architecture). The VAX floating point format matches for single-precision floating point numbers, but has two double-precision formats, neither of which follow the IEEE 754 standard. Reformatting of floating point values from and to the DIS standard is not necessarily an easy transformation. For the VAX, a conversion from one of the formats to the other and a simple multiplication of the result by four is needed. Other processors may require some difficult bit manipulation routines to perform the conversion.

Likewise, host processors vary widely in hardware network interface capability. The main concern here is in regard to conversion of the network physical medium to [illegible], thicknet, or AUI cable connectors. This conversion [illegible] straightforward, although costly under some [illegible]. Another network concern is the capability of [illegible] "Blue Book" Ethernet versus IEEE 802.3 protocols. Again, this is generally not a concern since interfaces can be switched between the two (Ethernet vs. 802.3), usually via a software protocol converter. But, PC network interfaces are generally Ethernet only.

At a higher level, implementation of DIS standards can be affected by programming languages and the standards for programming. For example, although the use of CASE tools is highly effective in code generation, real-time and object-oriented CASE systems are not available for PC assembler language (in which the pre-defined radar simulation was written). Likewise, for the case of non-standard floating point formats, the FORTRAN language may not be appropriate to implement the necessary bit manipulation for IEEE 754 conversions. This is usually not a major stumbling block, however.

Various coordinate system implementations were expected to be involved in the demonstration, because simulations would be running on varying hardware platforms simultaneously during during the networked demonstration sessions. For example, a simulation running on one host may use a world coordinate (Lat/Lon) positional system and another processor may be implementing a simulation using topocentric coordinates (X, Y, Z). These various implementations exist quite often in the flight simulation arena due to varying requirements for each simulation: one system may require a very large gaming area in which a geodetic coordinate system is necessary, whereas another system running in a smaller gaming area would only need a "flat earth" topocentric system. The DIS interface for these two examples would obviously be quite different in design, although hooks could be installed in the code to provide access to the correct coordinate conversion routines dependant upon the coordinate system locally used. So, regardless of which coordinate system is employed in a simulation, the DIS PDU packing/unpacking routines would convert to/from geocentric according to the requirements of the local simulator.

Databases, are utilized heavily in DIS implementations. Terrain, features, and moving models for visual and sensor systems all use these databases and they will use them differently, on differing processors, at differing levels of detail, and for differing reasons. Visual systems need highly detailed database inputs; low-level "fishing" radars need much less accurate database information. The correlation of these databases is required so that, say, a bridge appears at the same relative location on a visual IG as it does on a Digital Radar Landmass Simulation. Also, the correlation between the databases and entity positions and orientations must be accurate in order to keep ground vehicles from flying above the terrain or flying entities

from appearing to burrow through the terrain. This correlation is mainly a function of accuracy and consistency in coordinate transformations. If one high-detail system performs double-precision mathematics, its results will appear more realistic than one using lower precision algorithms. In fact, costly (in terms of CPU horsepower) floating point algorithms provide better correlation to databases. The choice of conversion algorithm may also affect correlation accuracy. If two hosts compute the same value differently, their position updates to the other hosts may not be well-correlated.

IMPLEMENTATION DETAILS AND DIFFICULTIES

In order to provide some more detail to the above-mentioned concerns, we address some realizations.

The FLY and radar simulations were designed to be efficient regarding network throughput. Assumptions have to be made in any simulation design, so that difficult requirements will fit within processing constraints. This means that corners must be cut (or at least shaved) in the design of DIS processes. Corner cutting must be selective, of course, in order to not reduce the realism of any simulation. Filtering of the PDUs is the most practical way to reduce both host network traffic and host resources. The filtering must be a bottoms up one; that is, the quickest and earliest filtering (or ignorance) of PDUs is essential. Since all PDUs are broadcast (in the current implementation), none could be ignored on network address alone, although that would have been the earliest filtering possible. Non-Ethernet packets could be ignored by low-level, raw interfaces, but VMS and UNIX do not afford this direct capability. However, filtering at this level was implemented on the PC. Also, non-UDP/IP packets fit the same mold as those of the non-Ethernet form. The demonstration used a single exercise ID, so filtering could not be performed at that level, either. Filtering finally can be implemented in any of the varying resources at the PDU type level. For example, the radar simulation was designed to ignore any PDU which was not an entity state, and the FLY routine ignored any PDUs not specifically designed for the demonstration (i.e., entity state, fire, detonation, and collision). The next stages of filtering can be performed on the types of entities provided within the PDUs. Thus, the radar simulation would ignore "small" entities such a dismounted infantry and light vehicles.

The radar simulation system was written using a flat earth topology. That is, the spherical earth was locally flattened (via a Sanson-Flamsteed transformation) into an X-Y coordinate system. This was dictated by the host flight simulation. Conversely, it was found to be most effective to design the FLY routine to operate in the geocentric coordinate system. Computing the DIS required geocentric positions was effortless and accurate, because flight models do not generate new positions, but rather generate coordinate system independent velocities.

This varying choice of coordinate system obviously injects some correlation errors. Reflectone's implementation was not too concerned with these errors, because the radar simulation was low in detail; so the issue remains open. Solutions remain to increase computational accuracy (double precision floating point evaluations) and to assure consistency in the choice of conversion algorithms.

Network I/O is much like radio transmissions: A radio transmitter is much easier to implement than a receiver, because there is no need to consider noise. Likewise, a network write command is much easier to design than a network read command because of all the extra (noisy?) packets and their asynchronicity in the receive mode. Implementation of the send/receive mechanisms depends upon the particular processor in use.

In practice, the implementation of network I/O on a PC can be handled in several ways. The most direct method involves proprietary software which is compatible with a single type of network interface card. This method was rejected as too costly in the long run because of being locked into a particular interface. As an alternative, there are public domain PC network drivers available from Clarkson University, the clearinghouse for a large set of consistent drivers. The consistency comes from a public domain specification for the drivers from a company named ftp Software. Each interface manufacturer writes the driver for his card and places the driver in the public domain. The interface to the drivers is assembly language – a perfect match for the radar simulation. The packet drivers are raw drivers – the user must perform bottom-level interfaces to the network's physical layer. For a PDU transmission, the user must place the appropriate Ethernet, Internet Protocol (IP), and User Datagram Protocol (UDP) headers and trailers around the DIS PDU and send this whole packet to the Clarkson driver interrupt with a "send message" command. For packet receipt, the following extended operations must be performed:

1. Locate the Clarkson driver and ensure it is installed,
2. Request device information from the driver,
3. "Access" the driver by supplying a receiver routine's address,
4. Initialize a packet-ready counter to zero,
5. On each pass through the simulation, check the packet-ready counter,

6. If non-zero, read the raw packet, and decrement the packet-ready counter. If zero, there are no packets ready and continue the simulation.
7. Check the packet for correct Ethernet headers,
8. Check whether the packet is an ARP request. If so, request the Ethernet address of the local interface card from the Clarkson driver, format an ARP reply packet, send the reply, and continue the simulation.
9. Check the packet for IP and UDP flags, and check the UDP port number for correctness.
10. Check the packet for smallest DIS PDU length,
11. Check the exercise ID and DIS Standard version for correct values,
12. Accept the packet as a DIS PDU.

ARP (Address Resolution Protocol) was required by the DIS Interoperability Demonstration. ARP involves a separate host requesting the Ethernet address of a known Internet node. The ARP request is transmitted in broadcast mode, is received and decoded by the appropriate node, and replied to by that node only. Thus, even though it was initially thought that the radar simulation running in the PC was to only receive packets, it was required to transmit an ARP reply also.

Most other implementations (including the costly PC version) involve direct "socket" library calls to send and receive DIS PDUs. Socket libraries handle the majority of the interface protocols from Ethernet, to IP, to UDP, including broadcast addressing capabilities. Socket send routines packetize the PDUs and receive functions remove the headers and trailers from packets returning only the PDU information. But, even these socket libraries require some extensive coding in practice. Once again, the send mode involves fewer steps than PDU receipt, but both involve the concept of binding a socket to a particular protocol (UDP/IP in this case) under a particular port number (DIS Interoperability Demonstration chose port number 3000.). Within the simulation loop, a "select" function determines if and when a probable PDU exists in the read buffer or when a socket is available for sending a PDU. We say "probable" PDU, because even with this higher level of coding at the socket library level the minimum PDU length, exercise ID, and DIS Standard version must still be checked in the returned packet to ensure that the packet is a PDU. Most implementations, including the VAX and UNIX implementations mentioned herein for the FLY routine, automatically check incoming packets for ARPs and reply as needed. With the use of socket libraries, however, comes a new set of obstacles. Broadcasting is a relatively high overhead network concept. Every broadcast packet must be received by every other node on the local area network. Thus each node's throughput is decreased because of the added packet count. Consequently, some systems (notably UNIX System V) disallow broadcast transmissions except by the "super-user." Only the root user, therefore, can operate on the DIS network. Broadcasting can clog a LAN very quickly with what is known as a "broadcast storm." These storms occur when routers amplify the number of broadcast packets to their local systems due to address servers re-sending ARP requests back to the broadcasters. This situation becomes unmanageable in wide-area networks.

Specific to the radar simulation, other design criteria concerned the landmass database and the placing of target returns on the database. PDU filtering was performed as described earlier, but with a twist. All current entities were displayed in a menu prior to operation of the simulation so that an ownship entity could be selected. This allowed the radar simulation to logically attach to any entity on the network and then display other entities within radar range and elevation. Concerns arise from even this basic selection. Filtering had to be done to another level, because the simulation allowed for only twelve targets (but there were many more displayable entities than twelve on-line at any given time). Adding to this complexity was the fact that entities tend to come and go during any demonstration. That is, some nodes on the network would bring in new entities asynchronously and others would drop off the network, especially during testing prior to the show. This was not taxing on the radar design, since lost entities simply would not be displayed any more, and new entities would be ignored. I counted a maximum of 212 entities at one point during testing which were actively filtered with little problem by the supposedly underpowered PC.

A larger problem existed when the ownship dropped off-line. The radar had to be redesigned drastically to alert to this condition. When the condition occurred, the redesigned system halted the radar display and went back to the "ownship selection mode" where the user needed to select a new ownship. Also, the algorithm for an entity dropping off the network remains rudimentary, because the DIS Standard lacks a protocol to indicate loss of an entity. This algorithm assumed that if any entity did not update itself for five seconds (the maximum time during which an active entity must send an entity state PDU as defined in the standard), then it is gone. This adds quite a bit of logic to any DIS implementation.

Another well-documented concern regarded dead reckoning (DR). In order to provide the most accurate DR, it was found that if everyone reckoned in the same coordinate system, less positional/attitudinal error was seen on displays. Not everyone on the demonstration network was dead reckoning in the same coordinate

system, although it was suggested that geocentric DR be used.

The radar's landmass database had to be preconditioned from SIF to one acceptable to the radar simulation. This is basically a local requirement for any simulation (e.g., visual vendors must pre-format the database to their specifications, etc.). Database preconditioning is a major DIS concern. The final SIF database was not available until late in the test sequence (August 15 for a November 1 use date), but preliminary databases were available. So a relatively minor reconditioning effort was all that was needed to install the new database. This brought about the design need for a general-purpose SIF conversion routine, so that when an updated SIF tape was delivered the routine could simply massage the data at the user's leisure.

SUMMARY/CONCLUSIONS

The use of something as conceptually simple as a "listen-only" radar simulation on a DIS network is much more involved than one could expect. Testing of a conceptual design is the most vital concern, one which requires at least one other host node to generate or receive DIS PDUs. Other concerns involve host capabilities (e.g., horsepower, math formats, networking capability), programming languages and design methodologies, and the use of Project 2851 SIF or other formatted terrain/feature/model databases. All of these criteria (or features) must be designed into any new DIS system, and should remain available for later use since the DIS Standard is an evolving document.

REFERENCES

Military Standard, "Internet Protocol Specification", MIL-STD-1777.

Military Standard (Final Draft), "Protocol Data Units for Entity Information and Entity Interaction in a Distributed Interactive Simulation", Institute for Simulation and Training, 1991.

Military Standard, "Standard Simulation Data Base Interchange Format for High Detail Input/Output and Distributed Processing", PRC, Inc.

Institute of Electrical And Electronic Engineers (IEEE) Standard, "Standard for Floating Point Numbers", IEEE 754 1985.

DISCLAIMERS

UNIX is a registered trademark of AT&T.

DEC, VMS, and VAX are trademarks of Digital Equipment Corporation.

Motorola and Delta are trademarks of Motorola, Inc.

Intel is a trademark of Intel Corporation.

PC is a registered trademark of International Business Machines, Inc.

CONCEPTUAL GRAPH ANALYSIS: A TOOL FOR CURRICULUM DEVELOPMENT, INSTRUCTIONAL DESIGN, AND TRAINEE EVALUATION

Richard T. Gill
Department of Mechanical Engineering
University of Idaho
Moscow, Idaho

Sallie E. Gordon
Department of Psychology
University of Idaho
Moscow, Idaho

ABSTRACT

The knowledge base in a given domain has an inherent structure within it, corresponding to the interrelationships between concepts, propositions, images, etc. Written language often makes that structure obscure because of its linear format and frequent ambiguities. Graphs can be used to make the structure of knowledge explicit, and this allows for a variety of knowledge engineering procedures and analyses to be performed with or upon the graphs. Conceptual graph structures are a particular type of graph, consisting of nodes and labeled arcs, which can be used to represent both declarative and procedural knowledge. The graphs rely on a highly specific syntax developed by Art Graesser and colleagues over a period of ten years, and have been empirically validated in several domains. The conceptual graph syntax described in this paper is a modified version developed at the University of Idaho specifically for knowledge engineering and instructional design purposes. Conceptual graph structures can be used to represent a variety of types of knowledge including taxonomic knowledge, goal structures with arcs corresponding to if-then rules, spatial knowledge, and causal knowledge. The structures can be used to depict the content and structure of a body of knowledge either for a particular individual or for a domain in general. This representational capability supports a variety of instructional activities including curriculum development and analysis, instructional design and development of instructional materials, and trainee evaluation.

ABOUT THE AUTHORS

Richard Gill is an Associate Professor of Mechanical Engineering at the University of Idaho. Dr. Gill received his Ph.D. in Mechanical Engineering (Human Factors specialization) from the University of Illinois, Champaign-Urbana in 1982. He conducts research in several areas of human factors including design of safety warnings and labels, development of tools for instructional design, and knowledge engineering for improvement of instructional text. Sallie Gordon is an Associate Professor of Psychology at the University of Idaho. Dr. Gordon holds a Ph.D. in Psychology from the University of Illinois, Champaign-Urbana, and conducts applied research in knowledge acquisition and instructional design. She has recently completed a book on the ergonomic design of training programs, to be published by Prentice Hall.

CONCEPTUAL GRAPH ANALYSIS: A TOOL FOR CURRICULUM DEVELOPMENT, INSTRUCTIONAL DESIGN, AND TRAINEE EVALUATION

Richard T. Gill
Department of Mechanical Engineering
University of Idaho
Moscow, Idaho

Sallie E. Gordon
Department of Psychology
University of Idaho
Moscow, Idaho

BACKGROUND

Many of the activities involved in training program development require the analysis of large bodies of knowledge. Several researchers in instructional design have suggested the use of graphs or knowledge networks to organize and evaluate such bodies of knowledge (e.g., Fisher, Faletti, Patterson, Thornton, Lipson, & Spring, 1990; Nordstrom & Clayton, 1988; Novak & Gowin, 1984; Jonassen, Beissner, & Yacci, 1993).

While this goal is worthwhile, efforts to use graphs for representing large bodies of knowledge in instructional design activities have been hampered by several problems. In the next section, we briefly review the more popular types of graphs, their use for instructional design and student evaluation, and the major drawbacks of efforts to date. In the remaining sections, we describe a graph method we are using in instructional design activities that alleviates these problems, some of our work to date, and the generic ways that the method can support training and instructional design.

Graph Types

Cognitive psychologists have suggested the use of networks to represent knowledge for over 20 years (e.g., Quillian, 1968). Most of these graphs have been a specific type knowledge, generically termed semantic networks. Semantic networks are graphs where nodes represent unitary concepts such as "bird," and the links represented relationships between them. The relationships in such networks usually convey taxonomic or descriptive relationships, such as Is-A or Has Property (see example illustrated in Figure 1).

Researchers in education have recently increased the use of semantic networks to represent individual or domain knowledge. In particular, Novak and Gowin (1984) suggested the use of "concept maps" for showing the interrelationships between concepts. Concept maps consist of concept nodes that are interrelated by various types of relationships. These relationships are unconstrained, and are typically whatever verbs are used in the text being studied or analyzed. The terms concept map and semantic network can be and are often used interchangeably. However, concept maps sometimes show causal relationships while semantic networks are usually restricted to taxonomic types of relationships.

A different type of graph is often used to describe and test psychological or instructional theories. These graphs are termed causal networks or causal models. In causal networks, nodes represent either explicit or implicit variables (such as "motivation"), and the links are of one type only, A Causes B. Although Jonassen et al. (1993) suggest their more general use for depicting causal relationships in a domain knowledge structure, causal networks have not been commonly used representational media in instructional activities.

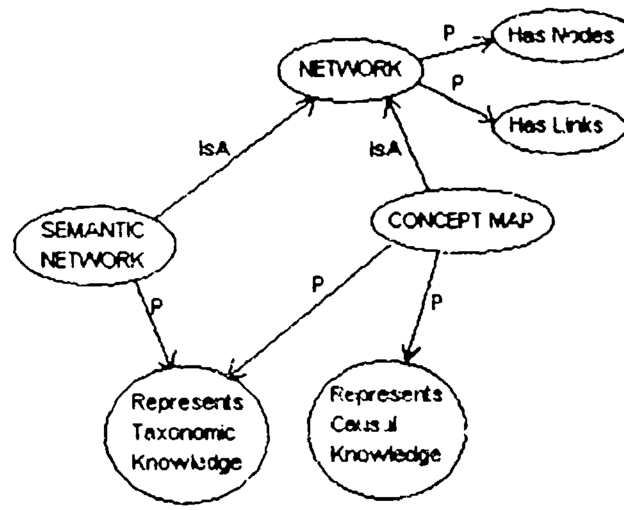


Figure 1. Semantic network showing taxonomic relationships. P denotes Has Property link.

Finally, Kieras (1988) has recently suggested the use of goal structures in performing task analysis. The goal structures have a specific format and syntax, known as GOMS, for Goal, Operators, Methods, and Selection Rules. These goal hierarchies are used to show the various means by which a person can operate some system to attain a higher order goal.

Instructional Uses of Graphs

Graphs such as semantic networks have been used for a variety of instructional design activities. Novak and Gowin (1984), and Jonassen et al. (1993) suggest using concept maps for planning entire curricula as well as more specific instructional activities. Nordstrom & Clayton (1988) have also suggested the use of concept maps to identify the interrelationships in children's literature and determine methods of teaching depending on individual learning styles.

Educational researchers are now arguing that concept maps can be useful instructional tools (Holley & Dansereau, 1984; Fisher et al., 1990). For example, Fisher and colleagues have students who are studying biology work individually or in groups to create networks of their knowledge as they acquire it. Ward (1988) proposes that construction of semantic maps helps learners build more complex knowledge structures and better identify the interrelationship among ideas.

Holley and Dansereau (1984) suggest use of the following types of relationships in supporting student learning:

X is an <u>Instance Of</u> Y
X is a <u>Property Of</u> Y
X is <u>Similar To</u> X
X is <u>Greater Than</u> (or Less Than) Y
X <u>Occurs Before</u> Y
X <u>Causes</u> Y
X is the <u>Negation Of</u> Y

Notice that most of these relationships are found in descriptive or taxonomic structures. That, is the concepts are used for definitions and descriptions of other concepts. Only the <u>Occurs Before</u> and <u>Causes</u> links provide a different type of knowledge, that of the interrelationships in a causal system. Such primarily taxonomic relationships and networks are commonly used in academic domains. One can speculate that this is because most of the information that is taught is general taxonomic knowledge.

Finally, some researchers suggest having the students graph their own semantic networks. These networks can then be used by the instructor to evaluate the accuracy of student knowledge (Fisher et al, 1990, Moreira, 1979).

Drawbacks and Difficulties

There are three major drawbacks to the graphical representation methods developed and used to date.

(1) Restrictive representation of knowledge types. First, the graph syntaxes used to date are too restrictive. That is, some graph types such as concept maps are useful for representing concepts and their taxonomic relationships. Others, such as GOMS, are useful for representing goal hierarchies. There is no graphing method used in instructional design that is capable of representing a wide variety of knowledge types, such as spatial knowledge, causal structures, goal hierarchies, images, if-then rules, etc. This type of representation is critical in real-world training because of the applied nature of the knowledge being taught.

(2) Lack of standardized syntax. A second problem with some representational methods is that there is no syntax or "language" per se. For example, concept maps rely on the use of verbs from text or whatever types of links that the researcher finds appropriate (reference). This lack of standardization creates several problems such as making it more difficult for two or more researchers to work together (no common language), difficult to compare different person's graphs on the same topic, and a proliferation of different types of links, many of which are more similar than different. This results in graphs that are poorly organized and "diffuse."

An exception to this problem is the syntax used in the GOMS method; this syntax is quite parsimonious.

(3) Lack of knowledge base management tools. Finally, until recently there have been no reasonable ways to manage the knowledge base if one moves beyond several dozen graph nodes. Graphs have simply been more difficult to manage because they were not amenable to manipulation on a computer.

In the previous six years, we have been using a particular graph syntax that overcomes the first two problems. More recently, we have been using a software program on a personal computer to store and analyze the graphs that is able to manage even very large networks. The graph syntax and the computer software are described in the next two sections.

CONCEPTUAL GRAPH STRUCTURES: THE REPRESENTATIONAL MEDIUM

Conceptual graph structures (CGSs) are a type of graph consisting of nodes linked by labeled, directional arcs.[1] Figure 2 shows a very small and incomplete example of a CGS, in this case depicting only taxonomic knowledge. It can be seen that each node contains specific content information and also a label specifying the type of knowledge. The information can be a unitary concept, or a statement. Statements are one of five node types:

Event
State
Style
Goal
Goal/Action

Any node in a CGS is labeled with one of these categories, either implicitly or explicitly (as in Figure 2). However, there are certain times when a researcher may want to leave the node labels out of the graphs (for example, when using them to structure interviews with subject matter experts).

As noted earlier, CGSs may contain different types of knowledge, such as taxonomic, goal hierarchies, causal structures, etc. These different types of knowledge are revealed by the types of arcs relating different nodes. For that reason, the arcs in CGSs are best described in reference to the type of knowledge they convey.

Table 1 shows the four major types of knowledge or substructures found in CGSs. It can be seen that taxonomic structures tend to predominantly consist of Is-A, Has Property (opposite of Property OF), and Has Instance arcs.

[1] Conceptual graph structures were originally developed by Art Graesser (Graesser & Clark, 1985); the syntax described in this article is that presented in Gordon & Gill, 1992.

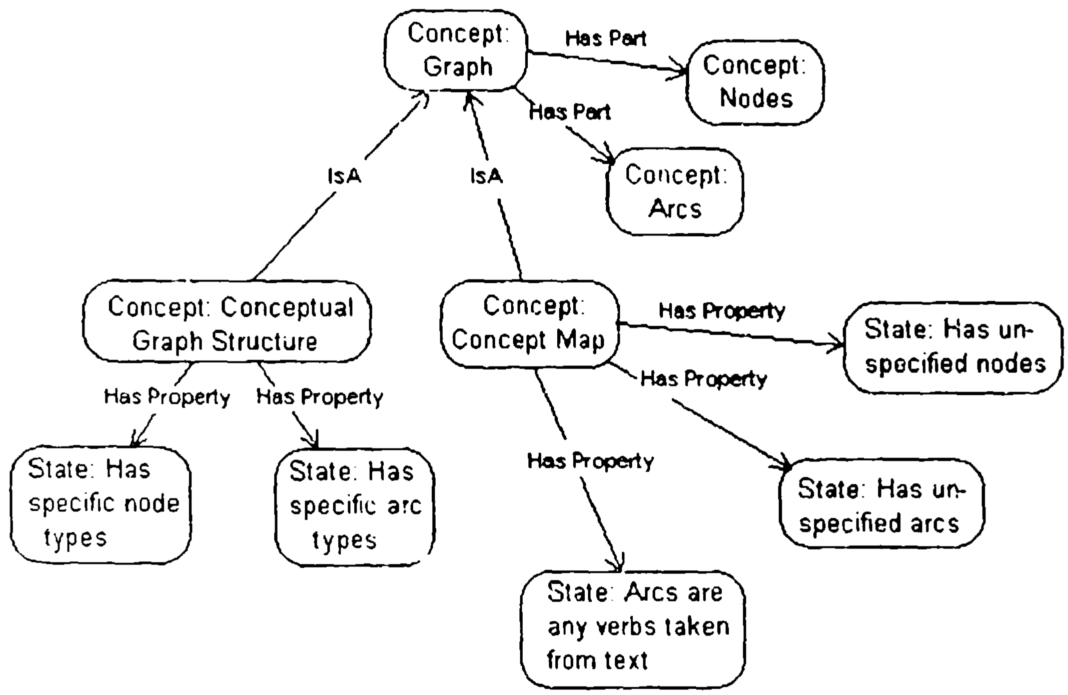


Figure 2. Conceptual graph structure showing taxonomic relationships.

Table 1. Conceptual graph substructures and arcs commonly used within the substructures.

TAXONOMIC STRUCTURES: Specify the relationships between superordinate and subordinate concepts (e.g., Apple Is-A Fruit).

- Is-A
- Has Property
- Has Instance
- Has Part
- Refers-to
- And/Or

SPATIAL STRUCTURES: Contain knowledge delineating the spatial layout of regions and objects in regions.

- Above/Below
- Left-of/Right-of
- Behind
- etc.

CAUSAL NETWORKS: Contain knowledge about causally driven state and event chains.

- Has Consequence
- Manner
- Before/During/After
- And/Or

GOAL HIERARCHIES: Specify goals, cognitive activities, and behavior procedures for accomplishing goals.

- Means
- Initiates
- Before/During/After
- Manner
- Has Consequence
- And/Or

It can also be seen that goal hierarchies predominantly consist of three types of links:

Means: A Goal/Action (goal or activity) is carried out by means of some activity

Initiates: A state or event initiates a particular goal/action

Has Consequence: A goal/action has some consequence

Causal structures contain mostly Has Consequence arcs (essentially the same as "Causes").

Figure 3 shows the graph for a small amount of information relevant to use of a VCR. It can be seen that several different types of knowledge or substructures can be contained or joined within one large graph.

Advantages of Using the Conceptual Graph Structure Syntax

Use of the conceptual graph structure syntax has several advantages over other graphing methods:

(1) Empirical support. The graph syntax and its use has received a wide variety of empirical support. For example: The conceptual graph

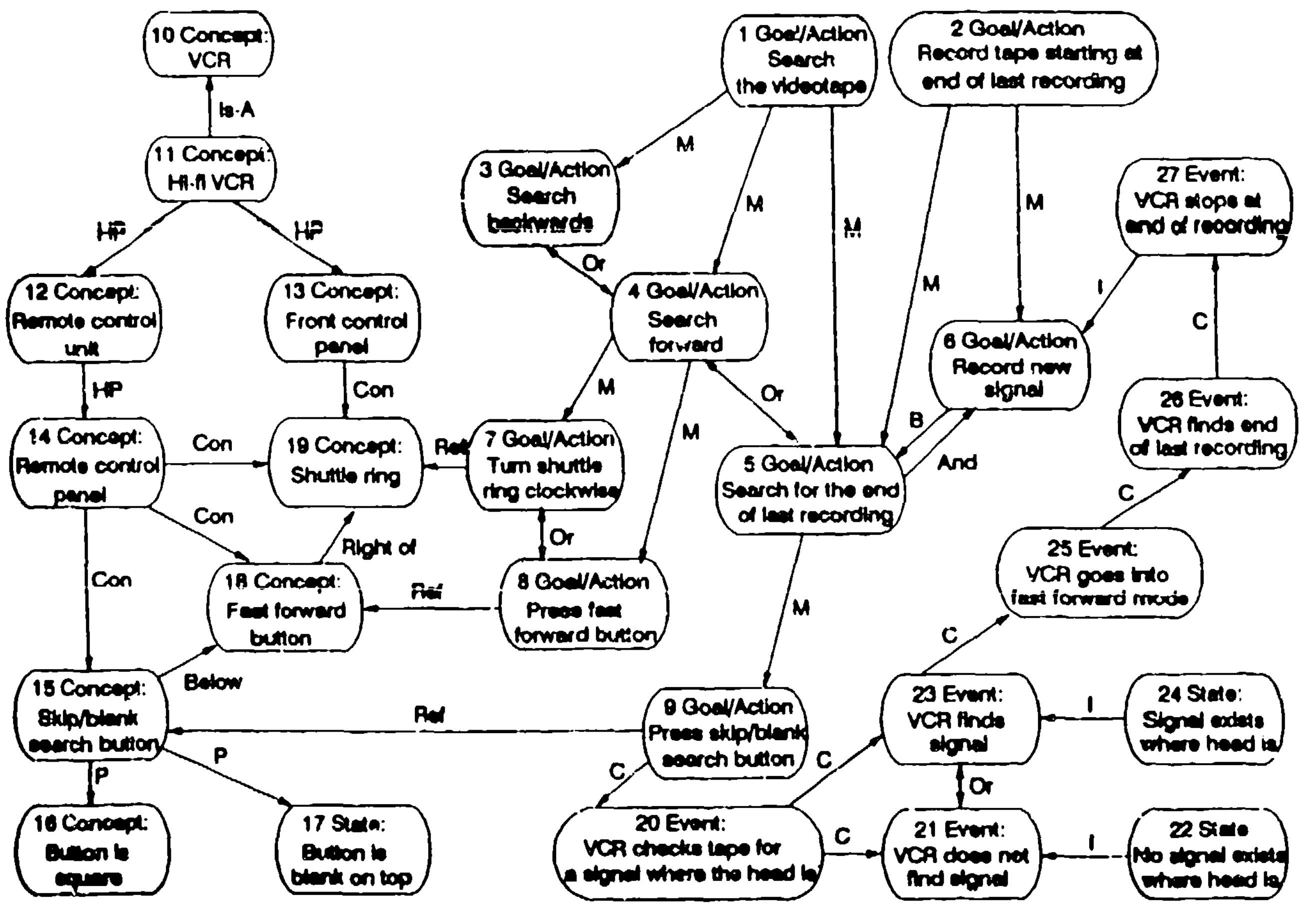


Figure 3. Conceptual graph structure for VCR with four types of substructure. Arc labels are: C: Consequence, Cont: Contains, HP: Has Part, I: Initiates, M: Means (by means of), and P: Has Property.

structure syntax itself has been shown to have high validity (Graesser & Franklin, 1990); the content and structure of individual's conceptual graph structures has been shown to predict subsequent problem solving (Gordon & Gill, 1989); and analysis of conceptual graph structures has been shown to increase the effectiveness of instructional text (Gordon, Schmierer, & Gill, in press).[2]

(2) Domain general. The graph syntax has successfully been used to represent knowledge in literally dozens of domains, ranging from narrative fairy tale stories to engineering mechanics (physics), forest stand prescription, use of VCRs and other mechanical systems, and procedures such as following recipes. This means that once a researcher or instructional designer learns the graph syntax (similar to learning any other language), that syntax can be reliably applied to new applications.

(3) Integrates different types of knowledge. Unlike other syntaxes such as GOMS or concept maps, the syntax can be used to represent and integrate all major types of knowledge including general taxonomic knowledge, goal structures, episodic knowledge, and visual images.

(4) Shorthand notation system. The graphs provide a useful shorthand for interviews, and a visual means to see interrelationships among concepts.

(5) Standardized syntax supports a systematic knowledge engineering methodology. The standardized node and arc syntax yielded and supports a method for knowledge acquisition, termed question probes (described below).

MANAGING THE GRAPHS ON A COMPUTER

Until recently, graphs of any size have been cumbersome to develop and manage. There are now several programs available for use on a personal computer to support this process. For most of our graph work, we are now using a special revision of a software product developed by the SemNet Research Group based in San Diego. The basic software is called "SemNet," runs on a Macintosh computer, and is available by writing to Kathleen Fisher at San Diego State University. The revised version for developing conceptual graph structures is known as "SemNet Wide." This is because the software was modified so that nodes could accommodate the longer text strings needed in some conceptual graph structure nodes.

The SemNet and SemNet Wide software supports quick and efficient development of graphs. Most people are able to develop graphs with the software with little to no use of support documents (truly a feat in this day and age). The graphs show a node in the center, surrounded by all nodes related by one link. Any of these surrounding nodes can be clicked, at which time that node becomes the center node on the screen. The nets can be traversed in this manner, or more quick "go to" commands are also available. The nets can be viewed in a variety of list formats, and selected parts can be viewed. The software can also provide helpful information such as the number of associations emanating from a given node (one measure of node "centrality").

CONCEPTUAL GRAPH ANALYSIS

Knowledge Acquisition

Development of conceptual graphs (CGSs) can be accomplished in a number of ways. In previous work, we have suggested a complementary set of knowledge acquisition methods for development of graphs (see Gordon et al., this conference, Gordon et al., in press; Gordon & Gill, 1992). These include Document Analysis, Interviews structured with Question Probes, Observation, and Rational Analysis. Document analysis is usually used for initializing the graphs. It consists of translating contents of relevant documents into conceptual graph form. A second way of initializing a graph is to ask a subject matter expert to briefly describe the domain or task of interest (Gordon & Gill, 1992).

Once the graphs have been initialized, they must be expanded and clarified. The most effective method for doing this is to use question probes with one or more subject matter experts (Gordon & Gill, 1992). Alternatively, if an expert is developing the

[2] This will be elaborated in a section to follow.

graph, the question probes can be implicitly asked of oneself.

Question probes are generic questions that are asked for each node on a graph. Each node type (e.g., concept) has certain types of questions. For example, an event node would result in questions regarding that event such as:

> What happens before _______?
> What happens after _______?
> What are the consequences of _________ occuring?
>
> Why does _______ occur?

Answers to the probes yield material to be added to the graph.

For procedural knowledge that is not easily obtained through interviews, direct observation can be used. Usually, an expert is observe and perhaps videotaped. Information such as initiating circumstances is identified and added to the graphs at appropriate points.

Finally, the graphs are evaluated using what might be termed "rational analysis." One main purpose for such an analysis is to identify any additional methods for accomplishing a task, or examples or concepts, etc. that should be included. For example, an expert may not have developed the optimal method for operating a system under all circumstances. An evaluation of the system itself might yield better goal hierarchies that those observed in actual performance.

Previous Applications of CGSs

We have used CGSs for several activities related to instructional design. First, the graphs have been used in several projects to perform cognitive task analysis (Gordon, in press). This results in a graphical representation of the knowledge that is to be contained in an instructional program.

Second, we have used the method to improve instructional text. For example, we graphed a portion of text from a major textbook in engineering mechanics. Subject matter experts were given question probes and the graph was "engineered" on the basis of answers to the probes. A new text based on this engineering process resulted in greatly improved problem solving by students (Gordon et al., in press). We have also used conceptual graphs to map out learner knowledge structures in detail. These individual graphs predicted problem solving with a hight degree of accuracy (Gordon & Gill, 1989). Finally, we are currently involved in development of an intelligent tutoring system based upon a task analysis using the conceptual graph analysis methodology (see Gordon et al., this conference).

GENERAL APPLICATIONS OF CONCEPTUAL GRAPH ANALYSIS

The purpose of this paper is to suggest the means by which conceptual graph analysis can be used to support a variety of instructional activities. In this section, we will briefly describe some ways in which this can be accomplished.

Curriculum Design

To determine the courses and course content needed to successfully cover a given domain, it is usually necessary to analyze a number of relevant documents and hold discussions with numerous subject matter experts. The resultant body of knowledge must somehow be organized, evaluated, and divided into courses and topics. Conceptual graph structures can be used for this purpose.

Several general approaches are possible. First, one might assign one individual to be responsible for creating a network using SemNet or some other appropriate program. This person could act as knowledge engineer and use interviews or questionnaires with experts. Alternatively, one could have several experts individually develop graphs and then merge the graphs into one. Finally, a group of experts could work together as a team (with a projection system) to develop a graph.

Normally, one would start by "free associating" with respect to the topics and skills needed for the curriculum plan under consideration. For example, if we were developing the curriculum for a Ph.D. in Human Factors, we would first obtain information relevant to any accreditation and licensure policies from all appropriate societies. We would use that information and

our own ideas to list all of the knowledge and skills necessary for practice in the human factors profession. These "nodes" could then be interrelated in a number of ways in the graph. The most obvious way is a topical hierarchy, with general topics subsuming more specific topics. Arcs used for this purpose could be Is-A arcs, or more specific arcs could be used such as Has Subtopic.

However, it is also a good idea to go through a look for other types of relationships to develop a richer network. For example, certain skills might be prerequisite for a number of advanced tasks. One can connect prerequisite skills to their goal/action nodes by Has Prerequisite arcs. By linking all of these prerequisite skills into goal hierarchies, a count can be taken indicating the "importance" of such lower level skills. This count can become the basis of a priority ranking for determination of remedial needs. SemNet can automatically provide such information.

Once the content has been defined and interrelated, the organization of the nodes can be evaluated and divided into coherent courses and topical outlines. The graphs generally fall into clusters of topics. SemNet gives a measure of "embeddedness" that can be used by the researchers to identify such clusters.

Instructional Design and Materials Development

Each section of a graph that pertains to one course or course section can become the basis for instructional design activities. For each topic, an instructor can add nodes describing instances or examples (Has Instance), analogies (Is Similar to), and learning activities (Means or Taught By Means of). The content material itself can be developed in detail and then converted into text, tutorials, or hypertext (Gordon, in press).

Trainee Evaluation

It is often advisable to evaluate the factual and procedural knowledge acquired by trainees, especially if there are critical consequences of errors in task performance. Identifying the structural knowledge of trainees is a good method for determining whether they have sufficient knowledge for task performance. Conceptual graphs have been shown to be highly sensitive measures of subjects' conceptual and procedural knowledge (Gill, Gordon, Moore and Barbera, 1988; Gordon & Gill, 1989). Question probes (Gordon & Gill, 1992) can be administered to trainees at appropriate times after learning activities. By evaluating how trainees interrelate concepts and procedures, one can determine exactly where there are gaps or misconceptions in the knowledge base (see Gordon & Gill, 1989 for a detailed description of methods for evaluating learner knowledge).

SUMMARY

The conceptual graph structure syntax can be applied to different domains with no or very few additions to the syntax. It can be used for a variety of instructional design activities, including curriculum development and analysis, and instructional design activities such as cognitive task analysis and development of instructional materials. It can also be used in conjunction with question probes to evaluate student understanding of complex topics and procedures, and to identify gaps and misconceptions in a student's knowledge base.

REFERENCES

Fisher, K. N., Saletti, J., Patterson, H., Thornton, R., Lipson, J., & Spring, C. (1990). Computer-based concept mapping. Journal of College Science Teaching, 19, 347-352.

Gill, R. T., Gordon, S. E., Moore, J., & Barbera, C. (1988). The role of knowledge structures in problem solving. Proceedings of the 1987 American Society for Engineering Education Annual Conference (Vol 2, pp. 583-590).

Gordon, S. E. (in press). Systematic training program design: Maximizing effectiveness and minimizing liability. Englewood Cliffs, NJ: Prentice Hall.

Gordon, S. E., & Gill, R. T. (1989). The formation and use of conceptual structures in problem-solving domains. Technical Report for the Air Force Office of Scientific Research, grant #AFOSR-88-0063.

Gordon, S. E., & Gill, R. T. (1992). Knowledge acquisition with question probes and conceptual graph structures. In T. Lauer, E. Peacock, and A. Graesser (Eds.), Questions and information systems (pp. 29-46). Hillsdale, NJ: Lawrence Erlbaum Associates.

Gordon, S. E., Schmierer, K. A., & Gill, R. T. (In press). Conceptual graph analysis: Knowledge acquisition for instructional system design. Human Factors.

Graesser, A. C., & Clark, L. C. (1985). Structures and procedures of implicit knowledge. Norwood, NJ: Ablex.

Graesser, A. C., & Franklin, S. P. (1990). QUEST: A cognitive model of question answering. Discourse Processes, 13, 279-304.

Graesser, A. C., & Gordon, S. E. (1991). Question answering and the organization of world knowledge. In G. Craik, A. Ortony, and W. Kessen (Eds.), Essays in honor of George Mandler (pp. 227-243). Hillsdale, NJ: Lawrence Erlbaum Associates.

Holley, C. D., & Dansereau, D. F. (1984). Spatial learning strategies: Techniques, applications, and related issues. San Diego, CA: Academic Press.

Jonassen, D. H., Beissner, K., & Yacci, M. (1993). Structural knowledge: Techniques for representing, conveying, and acquiring structural knowledge. Hillsdale, NJ: Lawrence Erlbaum Associates.

Kieras, D. E. (1988). Towards a practical GOMS model methodology for user interface design. In M. Helander (Ed.), Handbook of human-computer interaction. The Netherlands: Elsevier.

Moreira, M. (1979). Concept maps as tools for teaching. Journal of College Science Teaching, 8, 283-286.

Nordstrom, V., & Clayton, V. (1988). Book webbing using children's literature: A selective guide. References Services Review, 16(3), 73-80.

Novak, J. D., & Gowin, R. D. (1984). Learning how to learn. Cambridge: Cambridge University Press.

Quillian, M. R. (1968). Semantic memory. In M. Minsky (Ed.), Semantic information processing. Cambridge, MA: MIT Press.

COTS TOOLS FOR INTEGRATION AND TESTING OF MULTIPROCESSOR SIMULATION SOFTWARE

Margaret Opila Watkins
Harris Computer Systems Division
San Diego, California

ABSTRACT

The integration and troubleshooting of multiprocessor simulation software requires very specialized tools. They must be able to verify multiprocessor interaction and analyze time-critical areas of software. Historically these tools have been developed in-house for a specific application. But as multiprocessing development environments evolve, more tools are commercially available to assist engineers in the integration and testing phases. This paper discuses the features and usefulness of Commercial-Off-The-Shelf (COTS) versions of debugging and data monitoring tools for multiprocessor platforms.

ABOUT THE AUTHOR

Ms. Watkins is a Regional Support Specialist for Harris Computer Systems Division in San Diego, California. She has been providing consulting and product support to real-time simulation customers for the past eight years. She received her Bachelor of Science degree in Computer Science Engineering from the University of Illinois in 1984. In 1992, she received a Master of Business Administration degree from San Diego State University which included performing research in software development productivity.

COTS TOOLS FOR INTEGRATION AND TESTING OF MULTIPROCESSOR SIMULATION SOFTWARE

Margaret Opila Watkins
Harris Computer Systems Division
San Diego, California

INTRODUCTION

Multiprocessor computer systems have become more desirable as hosts for training systems for many reasons. This is due to influences from two sources: requirements growth and commercial availability. The demands upon trainers have increased because of the increasing complexity of systems being simulated, and the desire for them to be more true-to-life to improve the net transfer of training; resulting in more involved system requirements. The computer industry has advanced technologically, including the availability of industry standards and stable, multiprocessor systems. Computer hardware is now scalable to match requirements growth. Furthermore, the modularity of most simulation designs enables them to map logically onto multiple processors.

Design and coding of training systems for multiprocessors often proceeds smoothly. But upon integration and testing, advanced methods are required to assist in assurance of deterministic and repeatable performance and correct execution of multiple code paths. Timing issues such as sequential access to shared data, response to hardware interrupts, and the interference of debuggers with program interaction can make execution paths difficult to verify. Meeting hard real-time deadlines involves detecting and determining intermittent causes of execution overruns and contention overhead. Multiprocessing adds a additional layer of complexity for system integrators and testers. This may result in a need to control execution flow of some processes while stepping through another, or analyzing multiple streams of data produced by processes running in parallel.

A single human-friendly interface to monitoring multiple programs and assist in analyzing data collected is highly desirable. Intuitive easy-to-use tools, with sensible defaults and tailorable options, make the inherently complex task of multiprocessor integration simpler. Development of tools to assist during these later phases of software development is often more complex than developing the simulation software itself. Commercial-Off-The-Shelf (COTS) tools are as important in the integration and test phases as they are earlier on, and can also be leveraged to reduce the cost of the total project.

This paper will summarize some of the different categories of multiprocessor debugging tools commercially available and discuss how they are useful for specific situations commonly occurring in realistic training and simulation systems. The features of tools presented in this paper are representative of the products currently available to analyze multiprocessor interaction and small time-critical areas of execution.

For this discussion, the tools have been split into two general categories: multiprocessor debugging, and data monitoring.

MULTIPROCESSOR DEBUGGING

The main differences between a multiprocessor debugger and a traditional one are control of execution of multiple processes from a single point of human interaction, and visibility into the data areas of all processes. This saves the user from the effort of having to utilize various display devices and connect them to the simulation system.

There are many uses for a multiprocessor debugger. It can be used to set up and test scenarios of multiprocessor interaction, such as:

1) Test the execution sequence for boundary conditions.

2) Test the execution sequence for different combinations of process states.

3) Test combinations of execution sequences; e.g., what if this process reaches this point of execution before another process reaches a different point of execution? What if this is reversed?

4) Set up conditions to generate different input or output test files.

It can also be used for trouble shooting, such as:

1) Duplicate conditions in other processes leading up to a program fault and look for cause.

2) Step though programs and check for fault appearance at break points, in order to isolate the sequence of operations which caused it.

3) Pause the actions of all processes which may simultaneously attempt to modify a data structure, such as a linked list, while it is being examined in the debugger.

Finally, it is also a useful tool for controlling data monitoring, which is the second set of tools discussed in this paper. This requires a level of integration between the two types of tools.

The features available in multiprocessor debuggers which can help achieve the usefulness goals are explained in detail in the next section.

Controlling Execution

Besides the control functions of traditional debuggers: program start, breakpoint, step, and variable print; multiprocessor debuggers require additional features.

Process Startup - Support for different methods of process startup is important. Multiple process simulations may be started from shell scripts (job control macros), pipes or execs (such as a frequency based scheduler). Advanced debuggers have a modified shell, one in which every child process of it is automatically debugged. Sometimes, a user may not decide to start debugging until well into the training session. A command to attach the debugger to already executing processes can be used. If the process is already history, and has aborted, there is a method of recreating this history by looking at static process features such as the call stack and data values left in a memory dump.

Real-time Control - Once a program is started and in debug mode, there are many ways to effect its execution. Conditional breakpoints are useful for full-speed testing for failure. Combined with the use of state variables, complex conditional breakpoints can be used to check for a sequence of events which is suspected to occur leading up to a fault. Advanced debuggers implement this by having the capability to patch calls to user-written subprograms into the main process. The patch capability is also useful for situations where an engineer would like to try out a source change without having to recompile or restart the entire set of processes. Lastly, when combined with data monitoring tools, a form of the patch command can be used to set up instrumentation points.

The last method of process control is one where a halt is triggered from the keyboard, or mouse. When symptoms indicate an application fault or spinning process, the user can stop the program asynchronously and take a peek at what is going on.

It is important when the multiple processes are running, that they run near full speed in order to emulate true parallel processing interaction. This is achieved in many debuggers by actually compiling in code for conditional breakpoints, instrumentation points and subprogram calls. Also, in order to avoid the interference of the CPU usage of the debugger process, it can be isolated to an unused CPU.

Data Viewing - Multiprocessor debuggers must be able to not only look at local variables for each of the processes being probed, but must have access to global memory as well. Global memory is often shared among processes. It is also useful to be able to look at I/O memory on boards being integrated into the system.

Managing Complexity

In order to increase engineering productivity during integration of a suite of real-time processes, it is vital that the tools be intuitive to use, and have fast response. The human-machine interface must be designed so that the user can perceive and visualize the complexity of multiple events happening in parallel and be able to respond to each of them as quickly as reasoning occurs. Any feature which makes understanding the application or the tools easier will enhance the users productivity. These attributes are often the most complex and time consuming to integrate into home-grown

solutions. That is why the section on complexity management tools is included in this discussion.

Graphical Interface - Graphical interfaces are used in many ways to fulfill this mission. A logical mapping of processes to separate windows aids in visualizing process relationships. These windows usually include source views with events such as breakpoints highlighted, and program states clearly visible. Point-and-click push buttons for commonly used commands and cut-and-paste functions save time in thinking of the correct command syntax and limit typing errors. Displays of data structures and arrays with scroll bars make it easy to zoom in on variables of interest.

Getting Help - Documentation is critical to understanding the more complex features of a multiprocessing debugger. Examples and tutorials are a straightforward way to lower the learning curve of using a new tool. On-line documentation can be very helpful; with search capability, hypertext cross-referencing, and stacking of page views substituting for bulky manuals. Often there is context sensitive help linked to the specific screen which is active or the most recently occurring error.

Session Logs - Accurate logging of debugging events can be used as a "trail of bread crumbs" to verify sequences of execution. It is a significant time saver in tracking intermittent bugs, reproducing error conditions, building command macros and recording correct program behavior. It can eliminate the need to restart the debug session which may have gone on for hours with hundreds of interactive commands processed. Different types of information which can be logged are: the textual input and output of each process, commands typed into the debugger and debugger output such as data values and breakpoints reached. They can be sorted by process, or merged in order of occurrence with other processes to piece the history together. The logs can be saved on disk or retrieved temporarily by scrolling tools.

Miscellaneous Features - There are a few other features which are useful in certain situations. Language sensitive environments for conditions, function calls, assignments and patches can be useful for applications written in a combination of C, FORTRAN and Ada. Macros or scripts can save typing time and can be attached to a breakpoint to execute automatically upon occurrence. Commands can be grouped to apply to multiple processes at a time; for example, to cause a subset of processes to stop.

DATA MONITORING

Data monitoring is used to watch the changing of data values over the time of the simulation or training session. In addition to process variables, data monitoring can include items such as the process id, the CPU upon which execution occurs and a time-stamp. The advantage of data monitoring over viewing variables from a debugger is that the processes can be running at full speed without being stopped. It is language independent and data from multiple languages and multiple processes can be merged.

Data monitoring is useful for the following situations:

1) Verifying sequences of execution of parallel processes in real-time.

2) Verifying real-time determinism and worst case performance from a large sample base.

3) Identifying sources of interference causing processing overruns.

Its effectiveness depends upon both minimizing the interference of data collection with real-time processing, and turning the large volumes of data collected into information. Therefore, this discussion has been split into two parts, the collection of the data, and the analysis of it.

Data Collection

There are two types of information-gathering tools available today: sample-based and instrumentation-based. Each is useful in specific situations.

Sampling Method - A sample-based collection method is one in which data is grabbed from processes regardless of where they are in their execution path. It is usually done at a prescribed frequency. It is easily integrated with a live real-time display, as values can be updated on the screen each time they are probed. Variables to monitor are chosen on the fly by symbolic name; and programs do not have to be recompiled each time data preferences change.

Interference with real-time execution must be reduced by isolating the sampling process to its own CPU or else it will introduce a new source of indeterminism. This method is useful for getting a rough estimate of where problems occur relative to external hardware events. It can be used for instructor feedback on student activity. Also, if data is sent to disk, testbeds and data files can be generated.

This method has drawbacks if a fine grain of resolution is required for the data collected. For example, extreme occurrences of variable values may be lost if they happen between samplings. It is not possible to obtain the exact chronology of events with this method; synchronization problems can be missed and sources of process interference cannot be determined. Also, it is in[illegible] with this method whether variables [illegible] compared, as they may have been [illegible] different program states.

Instrumentation Method - Instrument-based data collection is a method in which data trace points are placed in the source code. The instrumentation of data collection at a specified point in code ensures that it is deterministic when the data is recorded relative to the execution sequence. The trace points can be added either to the source file and a recompile performed, or patched in with a compatible debugger. Some common areas for trace points to be placed in applications, operating systems, and Ada run-times are listed in the tables 1, 2, and 3 below. In the case of operating system trace points where source is not available, an instrumented kernel may be available from the vendor, and run as an alternate run-time system. If source is available, programmers must be careful not to break anything in timing-sensitive areas and reentrant sections. Each time a subsequent release of the operating system is used, trace points will have to be merged with the new source.

Since logging occurs at the exact instance of request, instrumentation allows a valid comparison of variable values in the same or adjacent trace calls. A time-stamp and process id can be associated with each piece of data in order to piece together a time-line of all processes of interest. Interaction among programs can be monitored by merging trace events. Critical areas, such as synchronization of access to shared data, or where kernel events occur during high priority process execution, can be "zoomed-in" on by adding multiple trace points. Timing data can be used for precise timing measurements. Interrupt routines cannot be stopped and browsed by most debuggers, but their variables and timing can be analyzed with a monitoring tool.

The main drawback of instrumentation-type data collection is that it requires modification of the application code. This can affect execution speed and program size, and correspondingly alter timing and addressing behavior of programs. There is a chance that this change may disguise the very problem one is trying to analyze. Just like the sampling method, it is important to minimize interference with application performance. A section of this paper is dedicated to how this is done.

TABLES OF TYPICAL INSTRUMENTATION POINTS

User Applications
• Suspected bug locations
• Exception processing
• Process, subprogram, or loop entry and exit points
• Branch locations
• Timing points, especially for clocking I/O processing
• Synchronization points of multiple processes
• Endpoints of atomic operations
• Endpoints of shared memory access code

Table 1

Operating System
• Interrupt entry, exit, nesting
• Exception entry/exit
• Context switch
• Page Fault
• System call entry/exit

Table 2

Ada Run-Time Tasking Information
• Rendezvous
• Delays
• Interrupts
• Signals
• Exceptions

Table 3

Limiting Execution Interference - There are many design issues that must be considered when building a data monitoring tool in order to minimize interference with the real-time application. Some of these are specific to a hardware architecture and require much research by the designer, making them too costly to develop in-house. They also prevent this type of tool from being very portable among architectures and operating systems.

There are quite a few tricks for reducing the overhead when recording an instrumentation point. Recording the data to a memory buffer in a raw format is the fastest method of saving information. It eliminates the unpredictability of an I/O operation. The memory buffer must be locked into memory so that a page fault does not occur, once again introducing an I/O operation. While data is being written to the buffer, all non-critical interrupts should be held off so that the return to application processing is as quick as possible. This also avoids reentrancy problems when trace points exist within interrupt processing routines. The speed of any synchronization constructs used in trace calls is also a consideration. The techniques for reducing the overhead of a tracepoint call also create determinism in the time it takes to record a tracepoint. The constant overhead can be measured and subtracted out of performance calculations.

Guaranteeing Accuracy - The accuracy of the data collected is crucial for fine-grain data recording. Time-stamps must be from a single, high-resolution clock which can be accessed quickly. A single timing source for all processors allows trace points from parallel processes to be merged into a historical time-line. A clock resolution of less than a microsecond can distinguish between most real-time events. The time to access the clock has proven to be the most time consuming portion of a trace event, so a hardware design that allows for quick access is essential.

Because multiple processes can attempt to record data simultaneously on a multiple processor system, reentrancy issues must be dealt with. If one process has written half its data, for example the CPU and process id, and another process gets the next memory access and puts in its variable values, the integrity of the data recorded will have been destroyed. A synchronization construct such as a spin lock should be used for protection of the buffer when in use. This introduces a source of inconsistency in recording overhead when multiple processes try to access the dump buffer at the same time and there are contention delays. This can be avoided by recording to separate buffers in the case of application logs, or minimized by limiting the number of instrumentation points used.

A buffer for data in memory has a size limitation. Contingencies must be available for when it fills up, because it will result in lost data. In some cases, the user may wish to have the buffer "wrap-around" and only save the most recent data available for analysis. If saving data for the full length of the simulation run, the user can be notified if data has been sent to the "bit-bucket." Then reconfiguration must be done; either increasing buffer size, reducing instrumentation calls, or dumping the buffer to disk more frequently; and the simulation rerun.

Saving Data - Data stored in memory is not usable by humans until it is either displayed on some device or copied to disk for later perusal. An instrumentation driver, which is separate from the applications dumping data, can be used to perform the copying function periodically. Since the instrumentation driver must perform I/O, its overhead and interference with real-time applications can be unpredictable. A spare or remote CPU can be used to retrieve the buffers. Users knowledgeable about the application can optionally control the timing of when the retrieve is done, so its impact upon application performance will be minimal. Code can be added to the application to trigger the dump when the execution is at an idle point. If there is no idle time available, or if only the most recent trace information is useful, as in the case of an intermittent bug; the user can let the buffer "wrap-around" continuously and dump data only after a fault is detected. The fault can be detected either by the application or by a human who then uses a keyboard input to trigger a data dump. The least predictable way to trigger data I/O is to have a threshold set on the buffer, and cause a data dump whenever it is reached.

Other types of control over the instrumentation driver are useful. These include determining the buffer size and number of buffers used. A user can also choose at run-time which tracepoints are to be enabled and recorded, without the need to recompile source or restart. Sources of user control over the instrumentation driver are: driver

startup options, shell-level commands or key strokes from the terminal passed to the running driver, triggers coded within the application, and commands from within an integrated debugger.

Data Analysis

Data analysis tools are what turn the bits and bytes collected into human-usable information. The ease at which this is done and the value of the information produced determine the utility of the tools. Discussion of analysis tools has been split into three topics: when the data is viewed, live or historically; computational analysis; and data graphing.

Live Viewing - The use of live data analysis was discussed briefly at the same time as sample-based data collection, as they are often used in combination. The same user interface used to display data values can be used to control the data collection, such as which variables to monitor and what frequency the sampling should occur at. There can be a switch to turn on disk recording so that the data can also be played back in historical mode. This method is useful for approximating the location of trouble areas. But it cannot be used for fine grain real-time analysis, due to the limited ability of humans to analyze detailed information before it flashes off the screen. It even has drawbacks in that the display and analysis add processing and I/O loading which may cause interference. Likewise live data analysis tools do not all possess the capability to "turn back the clock" and look at important areas that were missed previously.

Historical Playback - Historical data analysis occurs after all data collection is complete. The major advantage to this method is that the user has the time to analyze the data using more than one of the multiple methods discussed below. The user can also "zoom-in" on critical areas and spend more time analyzing them without missing out on the "future." Out of the massive amounts of data collected, only a small amount may be of interest. Sorting through these useless parts can take large computing resources which are best done off-line of the simulation. The drawback of this method is that once evidence of a trouble area is discovered, no changes can be made to effect that run of the simulation, such as collecting data for a different variable.

Computational Analysis - Computational analysis can be thought of as a filter through which the collected data passes and is turned into more useful data. Some types of beneficial filtering are:

1) Mapping raw data to symbolic names

2) Discarding types of events that are not of interest

3) Sorting events by time

4) Merging of multiple execution threads

5) Searching for an event or combination of conditions

6) Measuring time between events

7) Calculating summaries, such as average, highest, and lowest times and data values

There are some excellent user interfaces for the filtering tools available. Text tables can be used for mapping data into readable names. Strings for operating system and Ada run-time events are provided by the vendor. The sorting and merging are turned on by default and happen automatically. Search tools make use of the symbolic names assigned, and are menu driven with options for the direction of search and limiting the search to a certain interval. They can keep previous queries on a stack so that they can be recalled without retyping. Measuring the time distance between points can be done by picking the events off of a graphical view. The summary functions can be used in combination with the search tool. For example, to detect the cause of overruns: print the maximum execution time, use search to place the display interval of that event, then peruse the other events that are occurring near the same time.

Data Graphing - The graphing of data facilitates the visualization of changing data values and time gaps. It takes advantage of human intuition and pattern recognition in investigating application behavior. The integrated graphical display tools have convenient defaults which make them immediately usable, but are also customizable to feature specific data of interest and draw complex views when appropriate for the application at hand. Many COTS graphics tools

are available which can be set up and integrated with data collection. However most vendors supply a turnkey display system with sample screens included.

Modifying the Display - For intensive examination of data, an object-oriented editor can be used to define customized screens. It can be used to modify the sample screens or start from scratch. Display objects, states and constants can be defined. A pre-defined set of graphics objects make editing simpler. Some useful objects are:

1) Data graph of data value or expression vs. time

2) Event graph which is a tic mark indicating time of event occurrence with no associated data value

3) State graph displaying active periods beginning with one set of conditions and ending with another set

4) Ruler to mark increments which apply to the interval being displayed

5) Data box to display the alphanumeric string value of data

6) Text box for labeling and constant information

States are defined by specifying the starting and ending events and conditions. They are convenient to use in drawing state graphs, measuring durations, and specifying search expressions. Constants can be tables mapping data values to text strings which then can be referenced by the graphics and computational tools. Graphs created by the editor can be saved to and later restored from disk for use with data from a different run.

Interval Control - Once a graph is created, the entire display can be synchronized for all data being shown, and the activity of the processes can be viewed relative to time. Interval control mechanisms can be used to select the period of time for which data is displayed. They are: specifying time since start or event number, scrolling, zooming, and searching.

CONCLUSION

The goal of this paper is to make its reader more familiar with existing tools for testing and integration of multiprocessor applications. The characteristics of the tools available are dependent upon the target hardware vendor selected. The choice of tools offered is always evolving and current research must be done at the time of system selection to get the most up-to-date information.

The knowledge presented in this paper should be given weight in the selection of commercially available tools which will provide the real-time performance required and the lowest cost for the complete software life cycle. A checklist of features is described in Table 4. For each item, decision makers should ask themselves: How important is this feature? How do its benefits compare with its costs? How difficult would it be to develop in-house?

Features Checklist Table
• Performance overhead of tracepoint
• Clock resolution and speed of access
• Data collection control options
• Instrumented versions of OS and Ada run-time
• Live vs. post-mortem data analysis
• Ease in building display screens - sensible defaults plus tailorability
• Search and summary analysis tools
• Integration of debugger and data analysis tools
• Multi-language support
• Notification of dropped data
• Documentation, on-line help and support

Table 4

Future Directions

As multi-processing development tools evolve, it is expected that they will become more open and standardized. They will be useful not only with different platforms in a heterogeneous multiprocessing environment, but will work in conjunction with tools used at other phases in the software life cycle. Portability to other systems

will eliminate the learning curve necessary with vendor-specific tools. Standards evolving for real-time constructs, such as POSIX 1003.4, will assist in this evolution. This is also true for the Distributed Computing Environment standard which facilitates the control of processes running on remote systems.

REFERENCES

Baietto, J. and Leonard, W, "The NightStar Toolset," *Real-Time Magazine*, Real-Time Consult, Belgium, Q2

Plauger, P.J. 1993. "Enhancing Displays," *Embedded Systems Programming*, San Francisco: Miller Freeman, Inc., 89-92. May.

Stitt, M. 1992, *Debugging: Creative Techniques and Tools for Software Repair*, New York: John Wiley & Sons, Inc.

DEVELOPMENT OF A TECHNOLOGY FOR LANGUAGE INSTRUCTION USING MULTIMEDIA PCs

J. Peter Kincaid
Daniel Mullally
Catherine Meyer
William J. Bramble, Jr.

Institute for Simulation and Training
University of Central Florida

This paper describes the Language Technology Project which is now four years old, at the University of Central Florida's Institute for Simulation and Training (IST). The goal of this project has been to develop, evaluate, and commercially produce language courseware and training techniques for personal computers (PCs), including MS DOS (IBM-compatible) and Macintosh PCs equipped with voice interfaces. The project uses hypermedia software shells augmented by specially developed software for authoring the courseware. Researchers in the project are developing applications for computer-based language training and translation. These applications include language education for public school and university students, the "Forms Translator Assistant," "Dispatch" for rapidly teaching survival Spanish to 911 dispatchers, and "Survival Somali" developed for the U.S. Marine Corps for use in Somalia. Survival Somali was developed in five weeks from the initial identification of its need to delivery to the Marine Corps and is more fully described in another paper in this *Proceedings* (Mullally, Kincaid and Kishek). The ability to respond this rapidly was the result of resources and expertise already in place.

AUTHOR'S BIOGRAPHICAL SKETCH

J. Peter Kincaid
Dr. Kincaid is a Senior Scientist at IST. He earned his Ph.D. at Ohio State University and has over 25 years of human factors research experience including 15 years of university teaching experience. From 1979-1985 he was one of the Navy's lead scientist for developing English as a second language training courses. He developed the Department of Defense's readability standard. He has served as a NATO lecturer, and has worked for the Naval Training Systems Center, the Air Force Human Resources Laboratory, and the Army Research Institute. Dr. Kincaid has published over 70 journal research articles and technical reports. His current research activities include cost and training effectiveness, language technology, and human factors design of training devices with emphasis on computer-user interface.

Daniel E. Mullally Jr.
Mr. Mullally, a Research Associate at IST, has 20 years of military experience. As a training systems expert, he has served as a training consultant in industry, a war-games designer, and on several projects applying high technology to military training. Mr. Mullally is currently part of an R&D team designing semi-automated forces for a U.S. Army networked simulator. His research activities include the conceptual design and development of the Forms Translator Assistant for the U.S. Customs Service using a computer-assisted test and speech presentation methodology.

Catherine Meyer
Ms. Meyer has nine years of college and university teaching experience in French, Spanish and English as a second language. She is currently teaching foreign languages at the University of Central Florida, and is also an expert in instructional development, serving as the lead author and coordinator of production for the computer-based Spanish instructional aid "Pedro and Friends" (created at IST). Ms. Meyer is currently enrolled in the doctoral program in instructional technology, specializing in multicultural education, at UCF and is a Research Assistant at IST.

William J. Bramble Jr.
Mr. Bramble is a graduate of the University of Central Florida. He is currently enrolled in the human factors doctoral program at UCF and is also a Research Assistant at IST. He has been involved in the evaluation of computer-based language training aids and has presented his work at several national conferences. Mr. Bramble is also part of a research team developing human-computer interface for virtual environments.

INTRODUCTION

There are many potential benefits for developing computer assisted language learning (CALL) — (Kincaid, Mullally, and Kincaid, 1992). Interactive courseware which simulates an instructor who is a native speaker of the target language allows students more intensive individual practice.

The Language Technology project at the Institute for Simulation and Training (IST) has three areas of emphasis, to: (1) develop demonstration courseware, (2) evaluate the more promising demonstrations and conduct studies to establish effective learning strategies, and (3) commercially produce language courseware and training techniques.

Americans (including our military personnel) typically do not easily master fluency in foreign languages. Language books and audio tapes, used in schools, by industry and by the military Services are useful but have a number of shortcomings. The most important problem is that audio tapes make it awkward for the student to hear himself and then compare pronunciation against the recorded voice of the instructor. Students learning to speak a foreign language need extensive practice in pronouncing the language and receiving feedback. Ideally, this intense practice and feedback is provided by a native speaking instructor working with a single student. This is frequently not feasible and it certainly is expensive.

BASIC TECHNIQUE

Our computer-assisted language training software is based on simulating how a student is taught by a native speaker. For instance, an English speaker wanting to learn Spanish, listens to a native speaker pronounce words, phrases and sentences, which are also displayed. The student speaks into the computer's microphone, and the sounds are digitized and played back. The student's pronunciation can then be compared with the native speaker's. This cycle can be repeated as often as necessary. In another variation, conversations are simulated (e.g., the student converses with a bank teller) and two hours of intensive one-on-one instruction with the computer can be critiqued by an instructor in just a few minutes.

The voice interface is probably the most important element of integrated CALL. The voice interface allows the student to do more than simply listen and repeat. Until computer diagnosis (e.g. voice recognition) and feedback is developed to be effective and affordable to public schools, the built-in listen-record-and-compare feature is the best non-human phonetic production tutoring device available. There are at least two important applications for voice interface: training students to understand phonetic production (spoken language) and training them in phonetic production.

STRATEGY FOR THE R&D PROJECT

In our four year project, we have developed a number of strategies.

1. Make use of available low cost multimedia PCs and authoring languages.

Our courseware runs on both Macintosh and IBM-compatible PCs. Either of which, fully equipped to run our courseware, can be bought for under $1,000. For the Macintosh PC, our courseware runs on most old Macintosh PCs (such as the SE model) as well as all current models. The lowest priced of the current models costs less than $900. Macintosh PCs have a built in voice interface for their newest models which use the version 7.0 operating system. On older models, a voice interface, the MacRecorder (cost $75), plugs in.

The courseware works on IBM PC compatibles, models 286, 386 and 486. The lowest priced of the 386 computers equipped with a VGA color monitor, an 80 megabyte hard drive and a mouse can now be bought for about $700 and the price is still dropping. The voice interface we are using is the DigiSpeech unit (cost $150).

This research has been supported by grants from the Florida High Technology and Industry Council and the Florida Technological Research and Development Authority, through a contract with Rei America, Inc., and through cooperation with the U.S. Marine Corps and Apple Computer Corporation. The authors wish to acknowledge the contributions of the following individuals: Col. Bruce Schwanda, Roger Huder, George Kishek, Patrick Kelly, German Calderon, David Arnold, Xavier Calderon, Denise Kostic, Ana Aboudoj, Kim Telish, John Kincaid, Sheri Murtaugh, Susan Lanham and AbdiRizak-Salah.

We are using standard authoring software for producing our courseware. For the Macintosh, we are using either HyperCard or SuperCard. For the IBM PC courseware, we are using Linkway. We produce additional software modules, which we find necessary for all of our application programs, in C and C++.

2. Capability for Rapid Development.

Project personnel have followed a practice of quickly producing demonstration software, demonstrating a concept of computer-assisted language instruction in one to two weeks of concentrated effort. Demonstration courseware need not be a complete package; however, it should make apparent the value of full development.

One fully developed product, HELPS (Survival Somali) was developed in five weeks of concentrated effort (Mullally, et al., this volume). Given the extent of this project and the fact that it required the use of a language that the project team had no experience with, this was extremely rapid development. It was only possible, because the resources and personnel were already in place (and because of the availability of a Somali linguist).

3. Total Production Capabilities On-site.

One must have all resources in place for production of courseware including computers, authoring software, voice interface equipment, a complete graphics department, audio recording capabilities and the right mix of expertise. Key personnel on the project are expert in instructional design, language education and computer programming. Several are cross-trained, for example in language instruction and instructional design. This is a key to our capability for rapid development.

4. Formative and Summative Testing.

Testing, both to develop courseware and to formally evaluate the final version, is a routine part of the developmental process. Formative testing begins as soon as the earliest story boards have been implemented and provides continual feedback in the development process. Summative testing is conducted for courseware in the environment for which the courseware was developed. For example, "Pedro and Friends" was formally evaluated in elementary schools in Orlando, "Dispatch" in Orlando area fire departments and HELPS in Somali. The extent of the evaluation varies somewhat according to circumstances. For example, "Pedro and Friends" was subjected to a rigorous experimental evaluation comparing the performance of a control group against that of an experimental group. HELPS was evaluated using a questionnaire. Several of these evaluations are described in detail below.

5. Concentration on Specialized Applications.

Thje object is not to comprehensively teach a foreign language, but rather to concentrate on rapidly teaching limited (but useful) functional oral skills for particular applications. For example, HELPS is intended to provide the minimal survival language skills necessary for U.S. Marines to communicate relating to limited military and humanitarian topics. "Pedro" is a more comprehensive package, but it is designed to teach speaking and listening and does not stress reading and writing. "Pedro" is intended to supplement a traditional curriculum based on books and audio tapes.

6. Easy-to-Use Computer-User Interface

Each courseware project has been designed to be easy to use. For example, "Pedro" was tested with first grade elementary students, and without exception, they have been able to begin using the program in no more than five minutes. For a variety of our projects, a mouse and a windows-like environment are typically used to provide an easy to use interface. Ease of use is evaluated in the earliest phases of development and throughout the development process. In fact, a significant amount of development time and resources is devoted to developing an easy to use computer interface.

COURSEWARE DEVELOPED TO DATE

Courseware which has already been developed includes:

1. *Pedro and Friends/Pedro y sus Amigos.*

"Pedro and Friends" (shown in figure 1) was developed to assist second language learning at an elementary level. Versions are available for Macintosh, IBM compatible and Apple II PCs. The courseware consists of "talking" cartoon characters who take the student through various comic-book style "adventures."

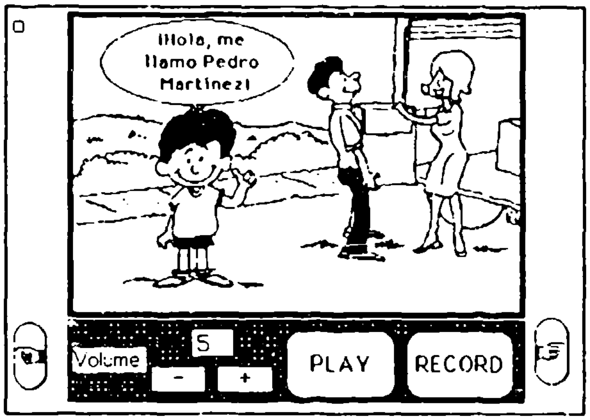


Figure 1. Screen From "Pedro y sus Amigos."

Also included in the software are drills and tests. The software is supplemented by a number of workbooks. This instruction is intended primarily for English as a second language instruction, but is equally applicable for foreign language instruction. Both English and Spanish voice tracks are incorporated into the program. The package is intended to supplement more traditional paper-based instructional material published by the same publisher, Rei America. The courseware has been evaluated in five elementary and middle schools in the Orange County (Florida) Public Schools.

2. *Advanced Conversational German* is an adaptation of conventional oral language instruction (consisting of a book and audio tapes) which is published by the German government. It was converted to interactive computer-based instruction and it was field tested in an advanced German class taught in an Orlando high school. Altogether, about 20 hours of intensive oral instruction were created. One interesting aspect of this project is that we are simply providing assistance to the high school which is doing most of most of the conversion work. Several talented high school seniors required only minimal training to do this work. Cost for this project is very low and it serves as a model for projects in other schools which tend to have limited funds.

3. The *Forms Translation Assistant (FTA)* (shown in figure 2) - was developed as a way to help non-English speakers fill out the various customs forms presented in their native language and provide additional information in their language.

The same technique is applicable for initial screening for military interrogations and for providing assistance to non-English speakers seeking emergency medical aid (for example in an emergency room).

The FTA resides on a standard IBM-PC (or compatible) computer with voice interface and a key pad. The computer is installed in a kiosk. Given the innovative nature of the approach employed in developing the FTA, many human-computer interface problems have had to be solved. For example, intuitive on-screen visual graphics and animation which match the associated oral instructions were developed. The FTA is currently implemented in English and Spanish. Training for use of the FTA has also been an important issue given that users: (1) speak a variety of native languages, and (2) may not be familiar with similar devices such as automated teller machines.

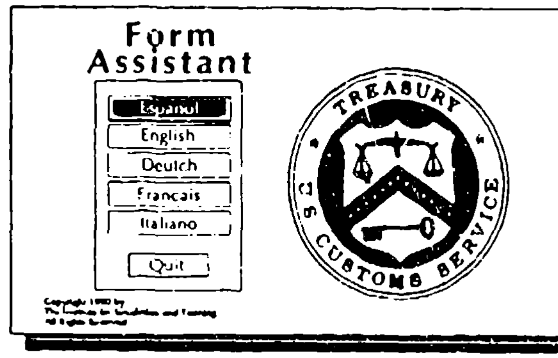


Figure 2. Opening Screen From "Forms Translator Assistant."

4. *Survival Somali* (also called HELPS) teaches the speaking of functional phrases and short sentences for U. S. military troops engaged in humanitarian support in Somalia. The program is hosted on a Macintosh PowerBook, which is a small portable PC with a built in voice interface. HELPS rapidly teaches key phrases to those who must have a functional grasp of the language in order to communicate orally. It also has a function to play pre-selected phrases through a public address system or through the computer itself. HELPS is described in detail in another paper of this proceedings (Mullally, et al.).

5. *Dispatch* teaches the minimal Spanish that an emergency, fire or police dispatcher needs to send the proper help. For example, the proper response for a domestic dispute is quite different than for a fire or a medical emergency. Also, the response for a child is frequently different than a response for an adult. Given the large number of immigrants who speak only Spanish, dispatchers frequently encounter this situation. The language data base for "Dispatch" is arranged as a hierarchy covering four different situations. The first statement in Spanish is

"I don't speak much Spanish, please answer yes or no." The typical dispatch can be made after four or five questions.

EVALUATIONS

Classroom Evaluation of "Pedro"

The most thorough evaluation we have conducted was for "Pedro" which was conducted at an Orange County (Florida) elementary school. The experiment compared the interactive computer version of "Pedro" with the same content presented as a comic book and audio tape.

Forty-six elementary school children were tested from fifth and sixth grade classes at Aloma Elementary School. They had no experience with Spanish. All subjects were native English speaking, and were proficient in reading and writing. Apple Macintosh computers were used to train the experimental subjects with the courseware. The recordings were made by native Spanish speakers.

Subjects were tested for their pronunciation of words and phrases with a definite "Spanish" sound. Native Spanish speaking judges scored each word for pronunciation accuracy and rating it on a seven point scale ranging from "unintelligible" to "native" quality.

Results suggest a strong positive learning effect for students utilizing the computerized practice method with voice recorder as opposed to the traditional technique. A t-test performed on the differences between overall pre-test and post-test scores for the two groups yielded a highly significant statistical result [$t(44) = 3.132$, $p = 0.003$]. From a practical standpoint, the experimental group using the computer presentation had an improvement 2.5 times greater than did the control group using the audio tape and book presentation. Moreover, the computer group displayed significant improvement after training ($p < 0.05$) for each of the 25 target words used to evaluate pronunciation performance. The control group, on the other hand, only showed significant improvement on 17 out of the 25 target words.

Forms Translator Assistant. Testing of the FTA was conducted in two parts including a pilot test using foreign students enrolled in the intensive English language program at the University of Central Florida, and a formal evaluation conducted at the Orlando International Airport.

Testing provided a successful proof-of-concept. Every subject tested at both UCF and the Orlando International Airport had an overall positive response to the FTA. Several revisions to the interface were made as a result of this testing. For example, the form was originally shown on two screens. This was necessary to be able to read the small print. Subjects had significant trouble in calling up the second screen (the bottom questions on the form) so a design change was made to enable users to navigate the entire form displayed on a single screen. Once the user selects an individual question, it is enlarged so that every word can be read.

Other results are: (1) the ergonomics design of the kiosk, which was based on military specifications, worked well with users having a full range of sizes, (2) the key pad worked well in accessing the program on screen (we used a "calculator," not a "telephone" key pad design), (3) the hierarchical arrangement of information (achieved by using a hypertext authoring system, Linkway) proved to be user-friendly, (4) a "286" IBM-compatible computer with a VGA monitor (either color or monochrome) is the minimum equipment configuration to host the FTA.

The FTA is clearly a viable concept which has been shown to do what it was designed for. The intuitive design features of the FTA interface lead to a rapid learning curve in achieving operator understanding the relationship between the key pad numbers and the on-screen superimposed numbering system. After a lengthy trip in an airplane the average non-English speaker arrives in the U. S. to the confusion and uncertainty of a confrontation with unfamiliar forms written in a strange language. The FTA can defuse a portion of that anxiety by providing personalized, interactive instruction with the rate of delivery determined by the user.

Dispatch. Testing of "Dispatch" is ongoing at the Orlando Fire Department, the Seminole County Fire Department and the University of Central Florida. Fifteen dispatchers at the Orlando Fire Department learned the required phrases in an average of six hours. An additional 15 dispatchers are being tested at Seminole County Fire Department. College students at the University of Central Florida are serving in a control group to compare learning using the computer presentation with learning using the same information presented in print and on an audio tape.

CONCLUSIONS AND DISCUSSION

What is the explanation for the consistent finding favoring the computer-based presentation over the conventional presentation of print materials and an audio tape? The answer is most likely the interactive nature of the computer program, and/or the computer's fast record-and-compare capability which sped up the practice process, and allowed the users to more objectively compare their own phonetic production with that of the computer's "native-speaker" voice. Further studies might isolate these factors to see how influential they are. Researchers might also like to examine the influence of this type of training on the learning of language comprehension. In any case, interactive listen-record-and-compare computer-based language training aids seem to give an increased effectiveness for oral language for the several applications we have tested so far.

REFERENCES

Bramble, W. J., & Hosley, D. L. (Eds.). (1991). *Improving foreign language teaching technology.* Research Report Developed for the Defense Language Institute, Foreign Language Center. Orlando, FL: Institute for Simulation and Training, University of Central Florida.

Garrett, N. (1991). Language pedagogy and effective technology use. In W. J. Bramble & D. Hosley (Eds.), *Improving foreign language teaching through technology.* (pp. 11-22). Research Report Developed for the Defense Language Institute, Foreign Language Center. Orlando, FL: Institute for Simulation and Training, University of Central Florida.

Kincaid, J. P., Meyer, C., et al. (in press). Pedro y Sus Amigos/Pedro and Friends: Level One. Produced by University of Central Florida, Institute for Simulation and Training. Miami, FL: Rei America.

Kincaid, J. P., Mullally, D., & Kincaid, J. F. (1992). Language training using computers with voice interface. *Educational Technology*, 90, 24-30.

DIS Network Traffic Analysis Estimation Techniques

Ken Doris
Grumman Aerospace and Electronics

Margaret Loper
University of Central Florida/Institute for Simulation and Training

Abstract

One of the most critical factors in developing Distributed Interactive Simulation (DIS) is that of defining the performance to be supplied by the network. Although the performance of a given network implementation is a scalable quantity, beyond a certain level (e.g., 1.5 Mbs) it becomes a heavy cost (and sometimes schedule) driver. The situation is further complicated by the interactive nature of DIS since each session or exercise poses a unique set of requirements. To the ability to accurately estimate those requirements in advance is a goal of the DIS community.

This paper will describe a mathematical estimating tool that is being jointly developed by Grumman and UCF/IST. A preliminary version of the tool, implemented in the form of a spreadsheet executable on either PC or Macintosh platforms, was used to estimate the network traffic prior to the DIS demonstration at the 1992 I/ITSC conference. A "post mortem" of that demonstration is included, comparing actual I/ITSC traffic to that predicted by this technique. The paper will also describe future enhancements to the tool

ABOUT THE AUTHORS

Kenneth Doris is a Technical Advisor in Grumman's Combat Systems organization. He is currently directing several research projects, including one devoted to DIS investigation. He is an active member of both the Communications Architecture and the Emissions subgroups of DIS. Last fall Mr. Doris lead the Grumman team at the 14th I/ITSEC DIS demonstration held in San Antonio. He holds a Bachelor of Electrical Engineering from Rensselaer Polytechnic Institute and has twenty-five years experience in Simulation and C^{3}I, specializing in computer architecture and software engineering.

Margaret Loper received a B.S. degree in Electrical Engineering from Clemson University in 1985 and an M.S. degree in Computer Science from the University of Central Florida in 1991. She is currently Principal Investigator for the Distributed Interactive Simulation (DIS) project at the Institute for Simulation and Training, Orlando. Her current research interests include simulation networking, OSI protocols, and multicast communications.

DIS Network Traffic Analysis Estimation Techniques

Ken Doris
Grumman Aerospace and Electronics

Margaret Loper
University of Central Florida/Institute for Simulation and Training

INTRODUCTION

One of the top objectives of Distributed Interactive Simulation (DIS) is to accommodate very large exercises. Just *how* large is typically described in terms of the number of entities that will simultaneously be active in the exercise. Numbers as high as 10^4 and 10^6 are being discussed as near term goals. While it is generally acknowledged that such exercises will generate a tremendous amount of network traffic, there is anticipation that technologies such as asynchronous transfer mode (ATM) and multicast addressing (MCA) will mitigate the problem. In any case, cost and availability of network resources will continue to be important issues to be considered in exercise planning. How much traffic will a give exercise generates is a common question. Or, looking at from the other viewpoint, given that the network has a specified maximum throughput, what size exercise can be accommodated?

The above questions have no straightforward answers since there is no direct conversion from exercise size to network traffic rate. Depending on the mixture of entity types (e.g., tank vs. aircraft) the types, sizes, and frequencies of messages (called Protocol Data Units or PDUs) varies. The situation is further complicated by the interactive nature of DIS -- The rate the PDUs are issued by each entity is a function of its relative activity at any point during the exercise.

NETWORK PERFORMANCE ISSUES

The network performance issues related to DIS are similar to classic network paradigms with the added requirement for real-time operation. This requirement places a premium on high throughput and low latency. Although the latter is of equal importance to the DIS community, this paper addresses only the throughput issue, specifically the development of a tool for estimating the DIS network traffic

DESCRIPTION OF A DIS TRAFFIC ESTIMATING TOOL

In early 1992, Grumman, as one of the authors of the Communications Architecture for DIS (CADIS) specification, developed a software tool to estimate DIS traffic. The tool uses a spreadsheet to calculate the network traffic in terms of bits per second, PDUs per second and packets per second (assuming PDUs are concatenated into datagram sized packets). It assumes that the vast majority of the traffic will result from (at most) the following subset of PDU types:

- Entity State PDU (ESPDU)
- Fire PDU (FPDU)
- Detonation PDU (DPDU)
- Emission PDU (EMPDU)
- Signal PDUs (Voice and Datalink)

It also assumes that each entity in a given exercise generates network traffic at a varying rate. The rate varies depending on the particular involvement of that entity with others. For example any vehicle that is transiting to or from its assigned duty area will exhibit very predictable dynamics and therefore generate low network traffic. Conversely, an entity entering into conflict or close cooperation with another will typically generate a fairly high level of traffic. In both cases the traffic is a result of the size and frequency of the PDUs generated. Estimating sizes of PDUs for selected entity types is a comparatively straightforward process while estimating the frequency at which they are generated is fairly complex and more subjective. The formulae for determining the size of the first four PDU types listed above are shown in Table 1

Given these formulae it is possible to estimate the PDU sizes for classes of entity types. For

PDU	FORMULA	REMARKS		
ESPDU	1152+128A	where:	A	= # of articulated part records
FPDU	704		H	= # of articulated parts hit
DPDU	800+128H		E	= # of emitters
EMPDU	192+E(160+B(416+64T))		B	= # of beams per emitter
			T	= # of targets in beam

Table 1 Formulae for PDU Sizes

example, for a given type of tank the minimum number of articulated part records may be 5 (azimuth and azimuth rate for turret, elevation othe barrel, and up/down position for two hatches) and the number of emitters 1 (laser range finder). Similar assumptions can be made regarding aircraft and surface ships (see Table 2).

The next step in estimating the bandwidth requirement of a given exercise is to approximate the rates at which the different entity classes will issue each of the above PDU types. Since this rate can vary a great deal within a given exercise, one method of estimation is to give values representing some average low and high rates (see Table 3).

Estimating the size and frequency of Signal PDUs does not fit into the above methodology. Each type of link will produce a variable sized PDU but at a constant burst rate. For example, a voice link will produce 65 Kbs (assuming 8-bit u-Law encoding) for as long as the speaker talks. At the low end, this could result in as few as 2 PDUs per second (assuming it is broken into maximum datagram sizes of 4K bytes) but this would cause severe intelligibility problems. A more likely number is 16 PDUs per second. Similarly, data links have fixed burst rates ranging from as low as 2.5 Kbs to a high of 230 Kbs and result in 4 to 16 PDUs/sec.

The final step is to determine the number of each major entity type and the tactical data links that will participate in the exercise. Given all of these factors, the determination of a range of probable network traffic for an exercise with a thousand entities, fifty voice channels and 20 data links can be easily calculated using the spreadsheet. (see Table 4).

APPLICATION OF THE TOOL TO THE 1992 I/ITSC DIS DEMONSTRATION

There are two types of simulation LANs homogeneous and heterogeneous. A homogeneous LAN is one in which all equipment (i.e., computing platforms, image generators, and simulation models) is supplied by a single vendor. For example, SIMNET constitutes a homogeneous LAN. Within this environment, processing delays and throughput capabilities are usually constant and predictable across all simulators.

Conversely, a heterogeneous LAN is composed of dissimilar computing platforms (PCs, workstations, etc.), image generators (fixed vs. dynamic priority) and simulation models. A heterogeneous environment introduces a range of operating speeds and performance to the network. One of the results of heterogeneity is a reduction in the number of entities that can be simultaneously represented on the network. An example of a heterogeneous LAN is the 1992 I/ITSEC demonstration network.

Because the I/ITSEC application of DIS was the first major implementation of a heterogeneous network, an analysis to determine the maximum number of entities that could be simultaneously represented was performed by IST several months before the actual demonstration.

The first step in the analysis was to define the simulator processing constraints. Five constraints were identified, as shown below.

1. the bandwidth of the physical media
2. the rate at which the physical interface hardware can read/write information (in PDUs/sec)
3. the rate at which data can traverse the communication protocol stack (in PDUs/sec)
4. the number of entities each simulator can track
5. the number of dynamic coordinate systems each simulator's image generator can manage.

	A	[illegible]	[illegible]	[illegible]	T	ESPDU	FPDU	DPDU	EMPDU
TANK	5	1	1	1	1	2220	1132	1356	1260
AIRCRAFT	10	2	3	1	2	2860	1132	1484	2732
SURFACE SHIP	50	5	10	1	5	7980	1132	1868	9580
						OVERHEAD BITS/PDU =			[illegible]

Table 2 Sample PDU Sizing

	LOW RATE				HIGH RATE			
	ESPDU	FPDU	DPDU	EMPDU	ESPDU	FPDU	DPDU	EMPDU
TANK	0.2	0	0	0.2	2	0.1	0.1	1
AIRCRAFT	0.2	0	0	0.2	8	0.1	0.1	4
SURFACE SHIP	0.2	0	0	0.2	1	0.1	0.1	4

Table 3 Sample PDU Rates By Entity Type

% ENTITIES AT HIGH RATE		0%	20%	40%	60%	80%	100%
% ENTITIES AT LOW RATE		100%	80%	60%	40%	20%	0%
# of TANKS	850 →	591,600	1,484,576	2,377,552	3,270,528	4,163,504	5,056,480
# of AIRCRAFT	140 →	156,576	1,079,210	2,001,843	2,924,477	3,847,110	4,769,744
# of SHIPS	10 →	35,120	121,296	207,472	293,648	379,824	466,000
# of TACTICAL VOICE LINKS	50 →	422,400	1,023,760	1,625,120	2,226,480	2,827,840	3,429,200
# of TACTICAL DATA LINKS	20 →	230,253	522,373	814,493	1,106,613	1,398,733	1,690,853
TOTAL TRAFFIC BITS/SEC	→	1,435,949	4,231,215	7,026,481	9,821,746	12,617,012	15,412,277
PDUs/SEC	→	580	1,616	2,652	3,688	4,724	5,760

Table 4 Sample Exercise Traffic Estimates

From a survey of I/ITSEC demonstration participants, values for constraints 2 through 5 above had a broad range (see Figure 1).

The next step in the analysis was to gather information on the maximum capabilities of each simulator participating in the demonstration. At the time of the survey, eighteen companies responded. These companies reported that they would bring a total of 26 manned/unmanned simulators and listen-only devices. This translated into a maximum total of 245 possible entities; many of the simulators such as CGF could represent multiple entities simultaneously.

From this list of entities, the Grumman DIS traffic estimating tool was used to characterize the network bandwidth. Inasmuch as participants did not anticipate exercising their simulator's maximum capabilities (i.e., producing the maximum number of entities) we chose to use a figure of 112 entities as the peak simultaneous load. (see Table 4). When comparing these predictions with the capabilities of the participants in Figure 1 the following observations

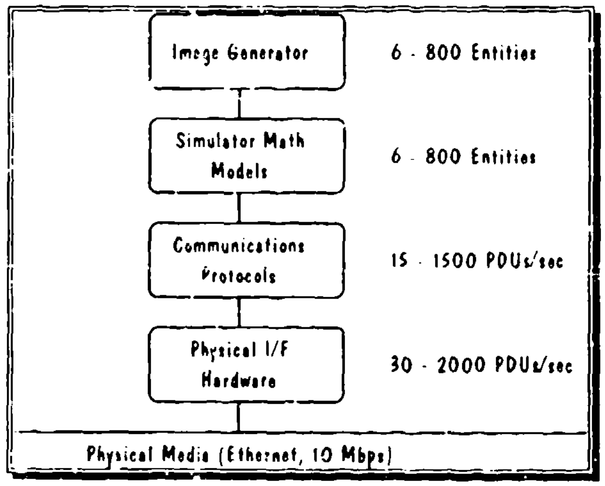


Figure 1 I/ITSEC Participant Performance Ranges

were made. While 731K bits/sec would not exceed the bandwidth of Ethernet, the rate of 311 PDUs/sec begins to push the upper bound of the processing capability of many of the simulators. For example, IST's PC-based simulators can process only 75 PDUs/sec at the Ethernet interface. Also, a 16 MIP single-processor

machine using commercial UDP/IP communication protocols and running only one process (i.e., receiving DIS PDUs but not running dead reckoning) can process only 200-250 PDUs/sec. When a second process is added that rate drops to 80-100 PDUs/sec. A rate of 311 PDUs/sec would quickly overwhelm both the Ethernet interface and the communications protocol processing of these simulators.

On the basis of this information, the I/ITSEC demonstration participants agreed to allow only the minimum number of entities on the network during the formal demonstrations. This prevented any adverse behavior during these periods. The actual number of entities running concurrently during the formal demonstrations ended up closer to 30; however, during non-demonstration times there were upwards of 100 entities simultaneously.

CONCLUSIONS

Obviously the DIS Traffic Estimating Tool has to be updated to account for the new PDUs in DIS 2.0. This could significantly increase the traffic loads seen by individual simulators. For example the newly proposed Emissions PDU may be generated as frequently as Entity State PDUs in some scenarios and the impact of Simulation Management on the network is still unknown.

The DIS Traffic Estimating Tool provided important information to the I/ITSEC demonstration participants for scenario generation. In the future, the estimating tool will be able to provide valuable insight on network bandwidth, allowing the DIS community to scale demonstrations and exercises as the capabilities of the simulators warrant.

During the conference week, a peak of 8 Mbps was observed when a number of jet aircraft began to dogfight and fire air-to-air missiles. This peak was not predicted by the estimating tool and was likely due to differences in dead reckoning implementations. We also observed that 80 entities on the network generated upwards of 350 packets/sec. This exceeded the predicted level of 311 PDUs/sec for 112 entities, as described earlier. This was most likely a result of non-DIS traffic on the network, such as Internet Control Message Protocol (ICMP) packets. Future versions of the tool should account for non-DIS traffic as well as inconsistencies in dead reckoning implementations.

REAL-TIME PDU FILTERING IN THE GATEWAYS OF WIDE AREA SIMULATION NETWORKS

M. A. Bassiouni and M. Chiu
Computer Science Department
University of Central Florida
Orlando, FL 32816

M. P. Garnsey
STRICOM
12350 Research Parkway
Orlando, FL 32826

ABSTRACT

Achieving the real-time linkage among multiple, geographically-distant, local area networks that support distributed interactive simulation (DIS) is one of the major technical challenges facing the implementation of future large-scale training systems. Data filtering is a technique that can help achieve this real-time linkage. In this paper, we present the results of a detailed study to design and evaluate the performance of data filtering for DIS systems. An approach suitable for the implementation of data filtering in the gateways of DIS networks is given and detailed performance results are presented. The paper is concluded by discussing methods to solve the problem of inaccurate state information at high filtering rates.

ABOUT THE AUTHORS

M. A. Bassiouni received his Ph.D. degree in Computer Science from the Pennsylvania State University in 1982. He is currently an Associate Professor of Computer Science at the University of Central Florida, Orlando. He has been actively involved in research on computer networks, distributed systems, concurrency control, and relational databases. Dr. Bassiouni is a member of IEEE and the IEEE Computer and Communications Societies, the Association for Computing Machinery, and the American Society for information Science.

M. Chiu received his masters degree in Computer Science from the University of Central Florida in 1991. He is currently a Ph.D. student and Research Assistant in the Department of Computer Science, UCF.

Mike Garnsey is a System Engineer with the U.S. Army Simulation Training and Instrumentation Command (STRICOM). He is currently working in the Combined Arms Technology Division on Distributed Interactive Simulation efforts such as the Army's Battlefield Distributed Simulation program. He holds a BS in Computer Engineering from the University of South Florida and a MS in Simulation Systems from the University of Central Florida.

REAL-TIME PDU FILTERING IN THE GATEWAYS OF WIDE AREA SIMULATION NETWORKS

M. A. Bassiouni and M. Chiu
Computer Science Department
University of Central Florida
Orlando, FL 32816

M. P. Garnsey
STRICOM
12350 Research Parkway
Orlando, FL 32826

INTRODUCTION

Today, there is a strong emphasis being placed on the development of efficient Distributed Interactive Simulation (DIS) systems[1]. DIS systems are composed of live, constructive, and virtual simulations interacting in a common synthetic or "virtual" battlefield from multiple geographically distributed locations via communications networks. Real-time data filtering and data compression are two complimentary techniques that can help improve the networking efficiency of DIS systems. The design of real-time compression for DIS protocol data units (PDUs) has been treated in a previous publication[2]. In this paper, we concentrate on the problem of real-time PDU filtering and present the results of a detailed performance evaluation study which we have recently conducted to assess the benefits of PDU filtering at the gateway level of wide area DIS networks.

PDU filtering refers to the process of analyzing the contents of communication messages in order to select only the data that is required to be received or transmitted by simulation entities. For example, if two simulated vehicles, say V_1 and V_2, are separated by a large distance in the simulated environment, then a state update message from vehicle V_1 would be irrelevant to (and would not therefore have to be delivered to) vehicle V_2. This example shows the most obvious method of filtering, namely, filtering based on distances in the simulated environment. Other factors (e.g., type of vehicles) can also affect the filtering process. For example, state update messages from a vehicle submersed in water could normally be ignored by vehicles on the ground. Filtering is used when the total traffic is large enough to overwhelm the small bandwidth of a local site or when the slow nodes in this site cannot handle the fast rate of message arrival. For example, if a high-speed FDDI backbone[3-5] is used to interconnect several 10 Mbits per second Ethernet simulation networks[6-7], filtering could then be used to reduce the size of the traffic flowing from the FDDI backbone to each individual Ethernet LAN. In large scale simulator-based training exercises, a simulated vehicle would normally need to receive information from only a small subset of the total simulated vehicles at any given time; state update messages from the rest of the vehicles would not be important and can be discarded.

PDU filtering at the WAN gateway level represents a level of abstraction of the Simulation Networking (SIMNET) program key design principle that all simulation entities be cognizant of each other. Basically, WAN gateway level PDU

filtering enables an efficient DIS system implementation principle that the cognizance of all simulation entities is not the responsibility of the underlying simulation entities of a LAN, but rather the responsibility of the WAN gateways. The WAN gateway must monitor and process all PDUs transmitted by other WAN gateways, and then relay only those PDUs of external entities that are within the "sphere of influence" of its underlying LAN entities to those entities.

AN APPROACH FOR DATA FILTERING

In this section, we shall give the high level details of an approach that can be used for implementing on-the-fly (i.e., real-time) filtering of state update messages. For the purpose of illustrating the basic ideas of the filtering scheme, we shall discuss methods relevant to simulators of ground vehicles and we shall use the distance separating these vehicles as the main criterion for filtering.

The filtering scheme uses a one-dimensional vector of distances for each simulated vehicle. The vector can be stored in the gateway of the LAN (cluster) where the simulator resides. Assuming that vehicles in the simulated environment are numbered 1 through n, the vector for the i^{th} simulator will be stored in the form

$$D_i = (d_{i1} , d_{i2} , \dots , d_{ii} = 0 , \dots , d_{in})$$

where d_{ij} is the distance (in the simulated environment) between vehicle V_i and vehicle V_j. For each vehicle, say vehicle V_i, we define a *"reachability region"* which specifies a neighborhood region such that the vehicles located within that region are tactically important to vehicle V_i (e.g., they are electronically visible to vehicle V_i). State update messages from vehicles outside this reachability region need not be delivered to vehicle V_i. For purposes of illustration, we shall assume that the simulated region is a flat terrain free from obstacles. In this case, the reachability region can be simply represented by a reachability radius R_i that gives the maximum distance from vehicle V_i at which another vehicle is reachable (visible). In addition to the distances vector D_i, a bit vector B_i is maintained for vehicle V_i and is defined by

$$B_i = (b_{i1} , b_{i2} , \dots , b_{ii} = 1 , \dots , b_{in})$$

where $b_{ij} = 1$ if $d_{ij} \leq s R_i$
$= 0$ otherwise

and s is a safety scale factor that suppresses the filtering of messages from vehicles that are outside the reachability region but which are close enough to its border. As shown in Figure 1, a safety ring of depth $(s-1)R_i$ is created to guard against any delay by the filtering mechanism in resuming the delivery of messages sent by a fast vehicle that suddenly entered the reachability region. Thus for example, if s is equal to 1.2, then vehicle V_i will start receiving messages from another vehicle even though that vehicle is at a distance 20% larger than the actual reachability radius.

B_i is a binary vector and is therefore more suitable than D_i for real-time filtering decisions. Upon receiving a state update message, say M_j, sent by vehicle V_j, the gateway will perform the following algorithm to update the vector B_i.

```
Update position of Vj based on Mj
for i = 1 to n and i ≠ j do
  if bij= 0 and dij ≤ sRi
  then bij = 1
  else if bij = 1 and dij > sRi
       then bij = 0; endif;
endfor
```

Because of the safety region, the above procedure does not represent a time critical computation; it can in fact be performed as a background job. Furthermore, even if the scale factor is set to 1 (i.e., no safety region), the above procedure can be easily implemented in real-

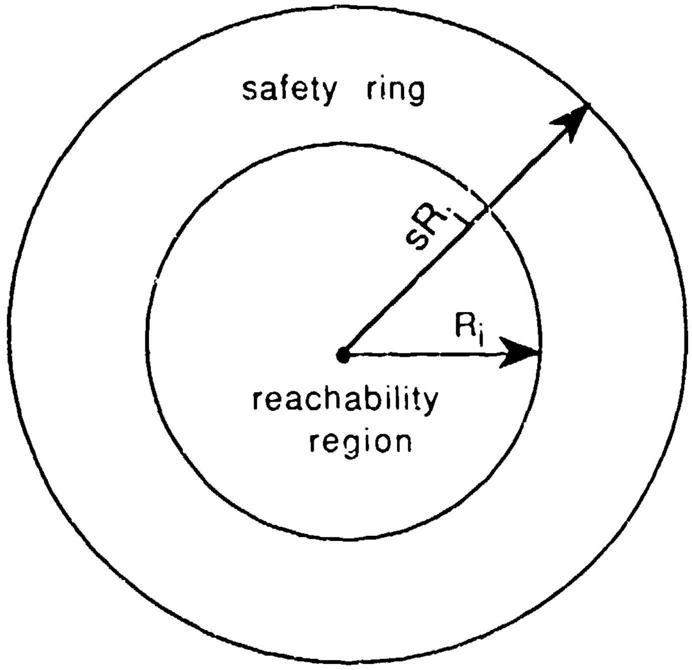


Figure 1. The reachability region

time using clever data structures or alternatively using parallel processing technology.

Using the above scheme, the filtering decision becomes an easy task. For example, to determine whether vehicle V_i needs to receive a message M_j sent by vehicle V_j, the following code is executed

if $b_{ij} = 1$ then send M_j to V_i
else discard M_j

IMPLEMENTATION OF DATA FILTERING

Filtering should be performed by network gateways at the transmission and reception of a message as well as during its routing in intermediate gateways. Filtering at transmission is the main process that could eliminate the majority of the unneeded messages. Filtering at reception performs a final check and could eliminate the unneeded messages that have not been detected during the transmission and routing phases. Notice that the gateway handles simulator messages in two different ways: 1) the gateway receives messages from nonlocal simulators (called external senders) and distributes them to the simulators on its local site, and 2) the gateway receives messages sent by the local simulators (called local senders) and transmits them over long-haul links to the simulators in other sites. The first case requires *filtering at reception* (i.e., filtering after receiving a message via long-haul links) and the second case requires *filtering at transmission* (i.e., filtering before transmitting a message onto long-haul links). We

shall start by discussing filtering at reception then proceed to examine filtering at transmission.

The receiving gateway would need to keep accurate information about the positions of the vehicles simulated by the local nodes connected to it. This can be done without much difficulty since the gateway receives every state update message transmitted by any node in its local site. Without loss of generality, let us assume that the total number of nodes (simulators) in all sites is n, and that the local site under our consideration contains the first m nodes, i.e., its nodes are numbered 1 through m. Using our filtering scheme, the gateway in this site maintains a collection of binary vectors equivalent to a binary matrix, called the filtering matrix B. In the case of filtering at reception, this matrix is defined as

$$B = [\, b_{ij} \,] \quad 1 \leq i \leq m, \; m+1 \leq j \leq n$$

where b_{ij} is a filtering flag that is set to 0 if messages from the external simulator j are not relevant (i.e., need not be delivered) to the local simulator i. As before, the safety scale factor is denoted by s and the reachability region of vehicle V_i is represented by a circle of radius R_i. Filter-at-transmission uses a logic similar to that used in the case of reception. The main idea can be briefly described as follows. If a local simulator sends a message, the gateway will perform filtering to transmit the message to only those external simulators that can be affected by it (or discard the message if it is not important to any external simulator).

PERFORMANCE RESULTS

A detailed simulation program written in Concurrent C was developed and used to evaluate the different design alternatives of filtering at transmission and filtering at reception. Several performance tests were conducted to assess the benefits of data filtering and compute the expected filtering rate in DIS networks. Fig. 2 gives the value of the overall filtering rate for various values of the reachability range and number of clusters (LANs). All the points shown in Fig. 2 use a total of 400 vehicles with a maximum vehicle's speed of 45 miles/hour. Fig. 3 shows the range of the filtering rate (low and high values) when 400 vehicles are distributed over different number of clusters (ranging from 4 to 25 clusters). An example plot showing the relationship between the reachability range and the overall filtering rate is shown in Fig. 4. The plot is for 80 vehicles distributed evenly among four clusters. Three cases of initial placement are tested in this experiment.

* Case A: vehicles in each cluster are initially grouped together and separated (in the simulated world) from other clusters.

* Case B: same as Case A, but one vehicle in each cluster is initially detached from its group and placed with some other cluster.

* Case C: All clusters initially overlap using random placement.

A similar plot for the relationship between the overall filtering rate and the number of clusters is given in Fig. 5. A reachability range of 50 miles is used in all the cases of this figure. In general, the filtering rate is more influenced by changes in the reachability range than by the number of clusters (using the same total number of vehicles).

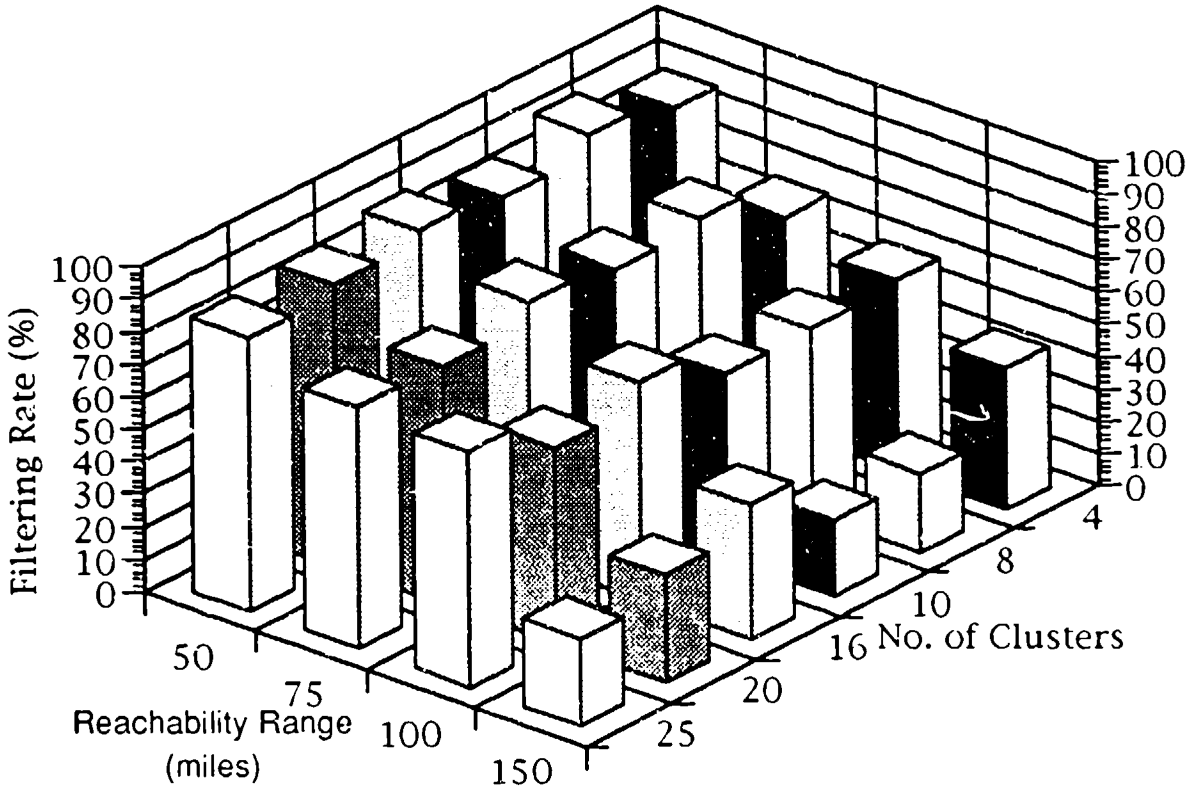


Fig. 2. Overall Filtering Rate For Various Values of Reachability Range and Number of Clusters

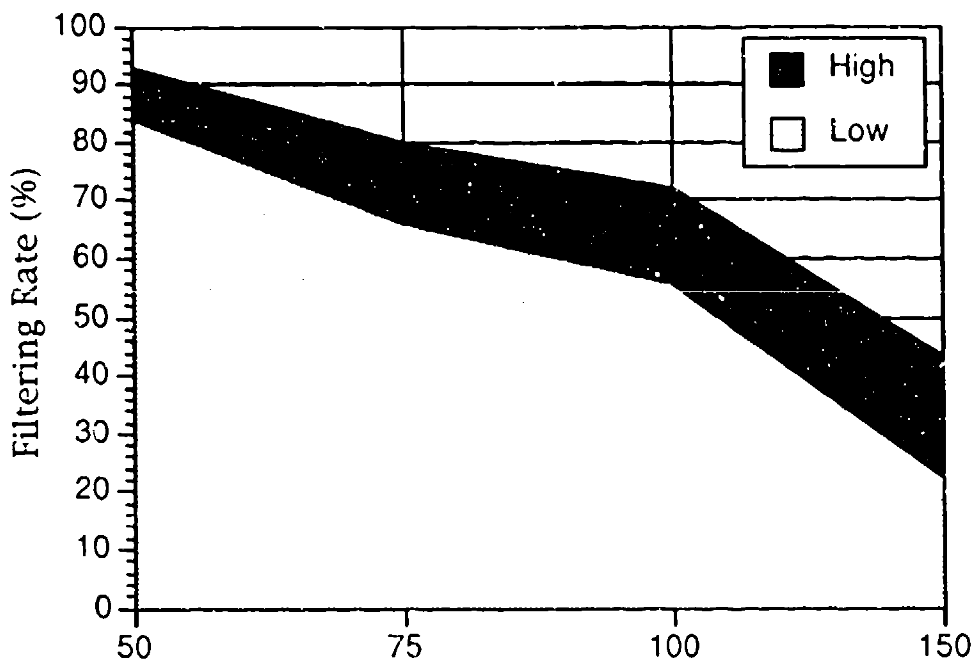


Fig. 3. Low and high values of filtering rate

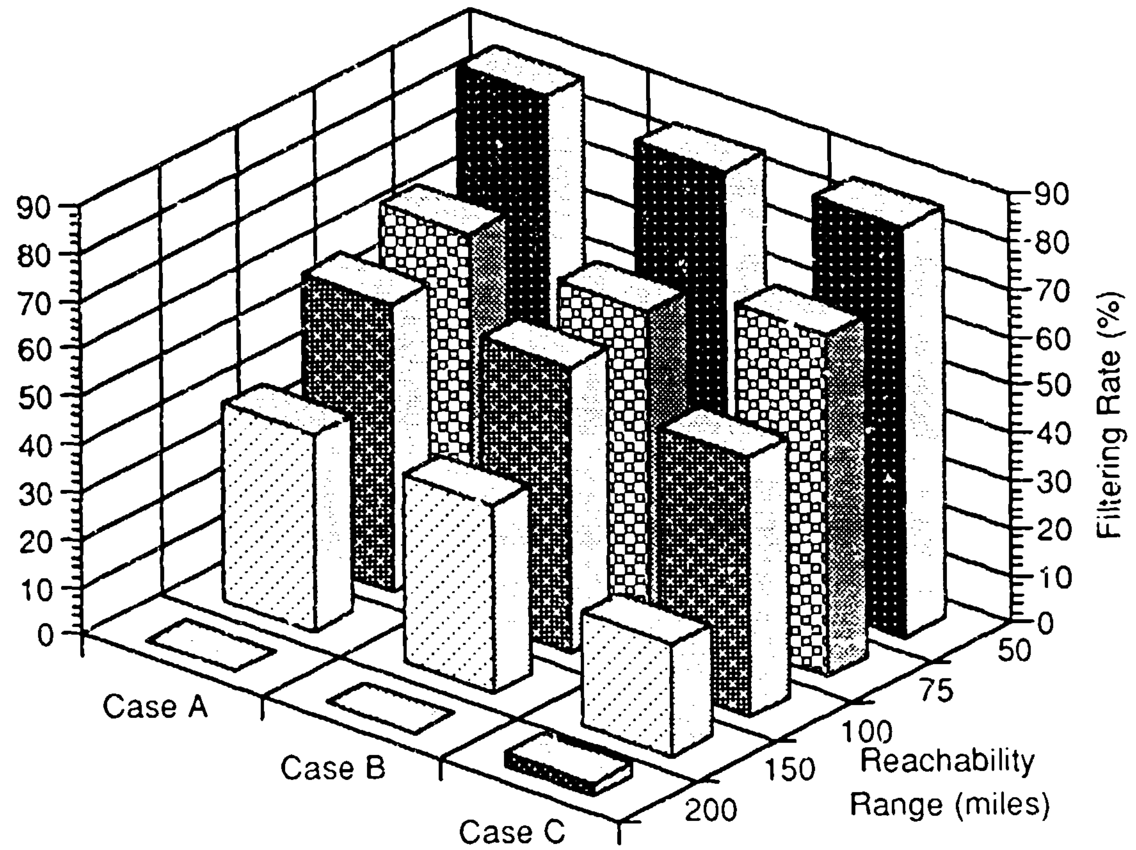


Fig. 4. Filtering Rate vs. Reachability Range

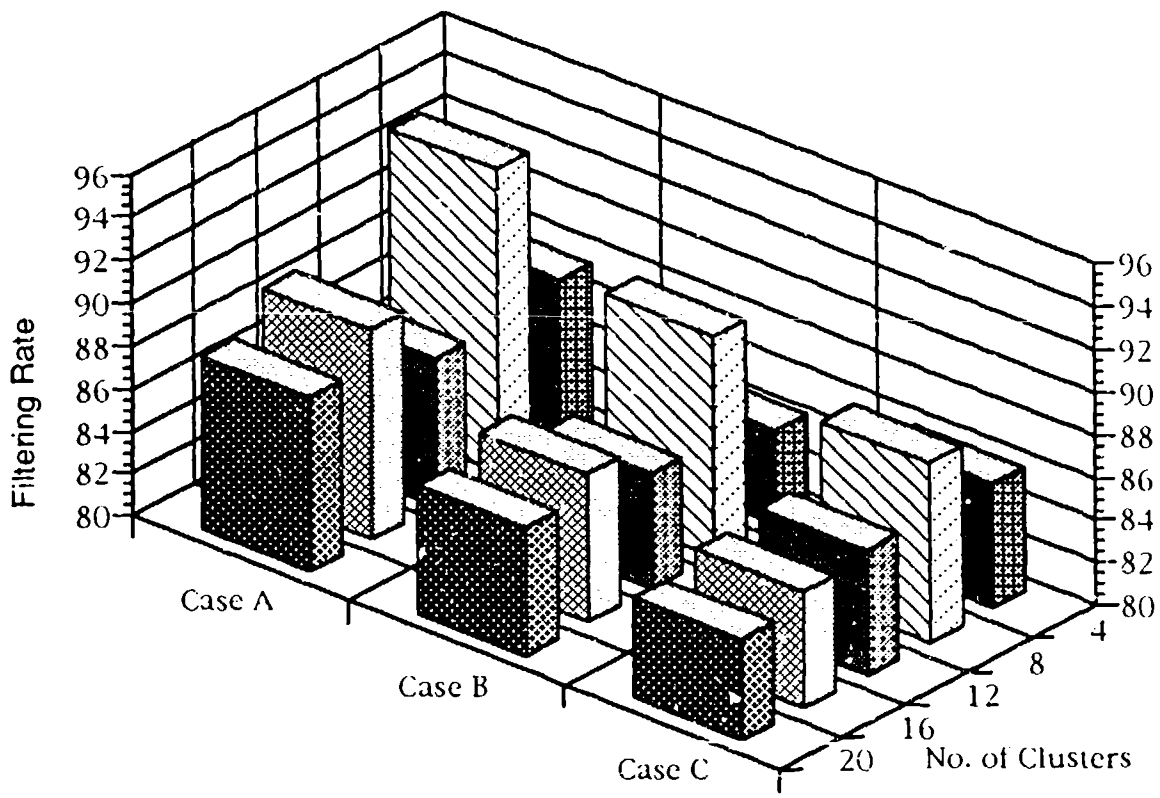


Fig. 5. Filtering rate vs. number of clusters

Fig. 6 shows the average, maximum, and minimum values of the overall filtering rate for the three cases discussed before. The total number of vehicles used in Fig. 6 is 80, the reachability range is 50 miles, and the number of clusters varies from 4 to 20.

DEAD-RECKONING AT THE GATEWAY LEVEL

If the filtering mechanism becomes very successful, the gateways will be deprived of receiving messages from some external simulators. This in turn will make the information (on external vehicles) maintained by each gateway less accurate and can render the filtering decisions incorrect.

A simple example will be used to illustrate this problem. Consider two vehicle simulators V_1 and V_2 located in two different DIS sites (LANs). The two sites communicate over long-haul links using the services of two gateways G1 and G2. Initially, the two vehicles are quite far from each other, i.e., each vehicle is outside the reachability region of the other vehicle.

Now assume that vehicle V_1 started moving towards vehicle V_2. Gateway G1 will execute the Filtering at Transmission strategy and will find that the state update messages emitted by V_1 need not be delivered to V_2. Gateway G1 will therefore refrain from sending these messages to G2. Thus this latter gateway continues to have the initial position of vehicle V_1. Now if vehicle V_2 moves towards V_1, gateway G2 may determine that the state update messages emitted by V_2 need not be delivered to V_1. G2 will therefore refrain from sending these messages to G1. The result is that G1 will have inaccurate information about the position of V_2. A situation can subsequently arise where the two vehicles V_1 and V_2 are near each other but each one of them is deprived of receiving the state update messages of the other.

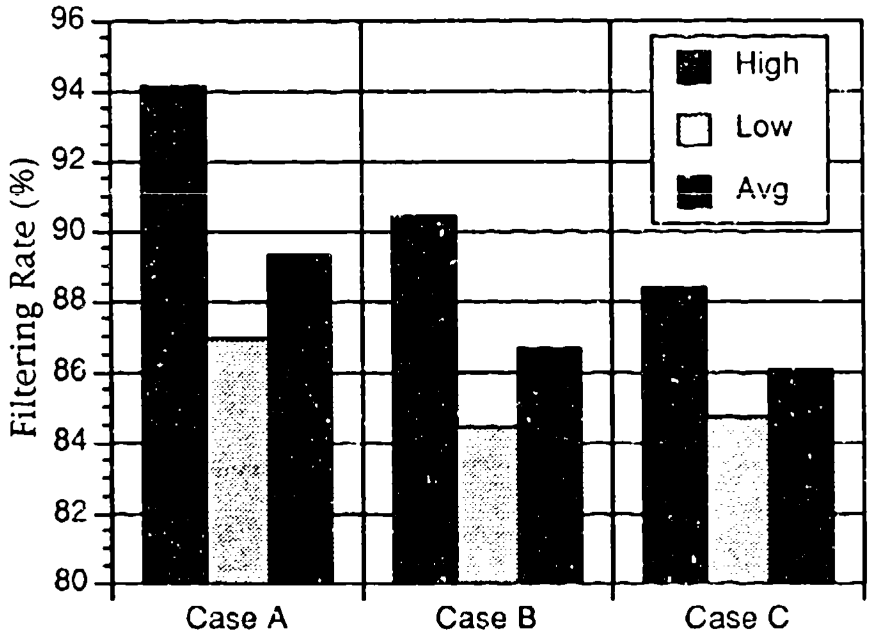


Fig. 6. Maximum, minimum, and average filtering rate

There are two ways to solve this problem. In the first method, each gateway would periodically suppress filtering for a short duration to ensure that the (simulated) positions of its local vehicles are recorded correctly in the other gateways. The second method is to provide the gateway with a **dead-reckoning algorithm** similar to that used by the vehicle simulators themselves. This approach is described next.

The concept of dead reckoning is used to reduce the number of state update messages that need to be transmitted by each simulator for the purpose of maintaining accurate state representation. Simply, each simulator has a high fidelity model which maintains accurate information (position, speed, velocity, etc.) about its own state. Each simulator also maintains a less accurate model, called the dead reckoning model, for each simulator (including itself) participating in the exercise. The dead reckoning model of a vehicle is periodically updated by extrapolating the information reported in the last state update message of that vehicle. Using first-order extrapolation, the anticipated position of a simulator is obtained by extrapolating its last reported position based on its last reported velocity as follows:

$$X(t+\tau) = X(t) + V_x(t)\,\tau$$
$$Y(t+\tau) = Y(t) + V_y(t)\,\tau$$
$$Z(t+\tau) = Z(t) + V_z(t)\,\tau$$

where $X(t)$, $Y(t)$, $Z(t)$ are the World Coordinates of the simulated vehicle at time t as reported in the last state update message, $V_x(t)$, $V_y(t)$, $V_z(t)$ are the x, y, z components of the velocity vector of the vehicle at time t, and $X(t+\tau)$, $Y(t+\tau)$, $Z(t+\tau)$ are the new coordinates predicted at τ units of time after the last state update message.

The prediction of the dead reckoning algorithm can be generally improved by resorting to higher order extrapolation equations. For example, the dead reckoning equations using second-order extrapolation are as follows

$$X(t+\tau) = X(t) + V_x(t)\,\tau + 0.5\,A_x(t)\,\tau^2$$
$$Y(t+\tau) = Y(t) + V_y(t)\,\tau + 0.5\,A_y(t)\,\tau^2$$
$$Z(t+\tau) = Z(t) + V_z(t)\,\tau + 0.5\,A_z(t)\,\tau^2$$

where $A_x(t)$, $A_y(t)$, $A_z(t)$ are the x, y, z components of the acceleration vector at time t. Whenever a state update message is received from a simulator, the information of that message is used to correct the extrapolated information of the dead reckoning model. Finally, when the state of a simulator actually changes, the simulator updates its own high fidelity model and compares it with the extrapolated information of its own dead reckoning model. If there is a large enough discrepancy between the two models, the simulator transmits a new state update message to all other simulators. The corresponding dead-reckoning approach in network gateways can now be described as follows:

1) Each gateway will maintain accurate information (position, speed, velocity, etc.) about each of the local simulators in its own site. This information (called the high fidelity model) should be reasonably accurate since the gateway receives every message transmitted by a local node.

2) Each gateway also maintains a less accurate model (called the dead reckoning model) for external simulators. The dead reckoning model is obtained by extrapolating the last reported location of each external vehicle based on its last reported velocity. Whenever a message is actually received from an external simulator, the information of that message is used to correct the extrapolated

information of the dead reckoning model.

3) Finally each gateway also keeps a dead reckoning model for its local simulators. When the gateway receives a message from a local simulator, it updates its high fidelity model and compares it with the extrapolated information of the dead reckoning model. If there is a large enough discrepancy between the positions of the local vehicle in the two models, the gateway transmits the message over the long-haul links.

Our tests so far have indicated that both methods discussed above are effective and give comparable results.

CONCLUSIONS

In this paper, we presented high level details of data filtering designs for DIS wide area networks. Our performance evaluation tests have shown that data filtering can provide a significant saving in the amount of traffic transmitted over the network. The paper presented various results showing the relationship between the overall filtering rate, number of clusters, and the reachability range. Gateway dead-reckoning methods to solve the problem of inaccurate state information at high filtering rates are presented.

ACKNOWLEDGMENT

This work has been supported by the Army's Simulation, Training, and Instrumentation Command (STRICOM) under contract N61339-92-C-0016.

REFERENCES

[1] LORAL System Company. Strawman DIS Architecture Document. Vol. I: Summary Description. ADST Program Office, March 1992.

[2] Bassiouni, M. and Mukherjee, A. "Data compression in real-time distributed systems" Proc. of IEEE Global Telecommunications Conference, 1990, pp. 967-971.

[3] ANSI Standard X3.148 "FDDI token-ring physical layer protocol (PHY)" 1988.

[4] ANSI Standard X3.139 "FDDI token-ring media access control (MAC)" American National Standard, 1987.

[5] Bassiouni, M. and Thompson, J. "Application of FDDI/XTP network protocols to distributed simulation" Proc. of the 12th I/ITS Conference, 1990, pp. 226-233.

[6] IEEE/ANSI Standard 8802/3 "Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specification" IEEE Computer Society Press, 1985.

[7] Bassiouni, M.; Georgiopoulos, M. and Thompson, J. "Real time simulation networking: network modeling and protocol alternatives" Proceedings of the 11th (I/ITSC) Conference, 1989, pp. 52-61.

[8] STRICOM *"Ditsributed Interactive Simulation", Standards Developement Draft 2.0*, September 1992.

INDEX BY AUTHOR

INDEX BY PAPER TITLE

NOTES